CONCEPTUAL DESIGN TOOL FOR FUEL-CELL POWERED MICRO AIR VEHICLES

THESIS

Paul M. Hrad, Captain, USAF

AFIT/GAE/ENY/10-M12

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GAE/ENY/10-M12

CONCEPTUAL DESIGN TOOL FOR FUEL-CELL POWERED MICRO AIR VEHICLES

THESIS

Presented to the Faculty

Department of Aeronautics and Astronautics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Aeronautical Engineering

Paul M. Hrad

Captain, USAF

March 2010

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

AFIT/GAE/ENY/10-M12

CONCEPTUAL DESIGN TOOL FOR FUEL-CELL POWERED MICRO AIR VEHICLES

Paul M. Hrad

Captain, USAF

Approved:

//Signed//	16MAR10
Frederick G. Harmon, Lt Col, USAF (Chairman)	Date
//Signed//	16MAR10
Christopher D. Shearer, Lt Col, USAF (Member)	Date
//Signed//	16MAR10
Dr. Mark F. Reeder (Member)	Date

AFIT/GAE/ENY/10-M12

Abstract

This research proposes a conceptual design tool to determine optimal performance requirements of a hybrid-power system for Micro Air Vehicles (MAVs) comparable in size to the Cooper's Hawk. An inviscid aerodynamic code, Athena Vortex Lattice (AVL), and a motor-propeller analysis code, QPROP, provide overall lift, drag, and thrust data for power-required calculation as functions of many variables to include mass, platform geometry, altitude, velocity, and mission duration. Phoenix Technologies' Model Center was used to integrate these multi-disciplinary components with a power management algorithm that can determine propulsion system mass for a given specific power and specific energy in order to perform the requested mission. Any component or subcomponent can be analyzed within Model Center using trade studies and optimization. As additional analysis software becomes available, it will be easily interchangeable within the Model Center architecture. This proof-of-concept study demonstrates a high-level analysis tool for MAV propulsion system design and improvement. The tool was used to simulate an Intelligence, Surveillance, and Reconnaissance (ISR) mission for the fixed wing Generic Micro Aerial Vehicle (GenMAV) and determined system level power and hybrid-power source requirements. The sizing strategy between a high specific power source and a high specific energy source was investigated. Due to the current technology of small fuel cells with low specific power, results show that a MAV fuel cell-battery hybrid-power system would not perform better than a pure battery or battery-battery power system. However, a feasible fuel cell capability requirement of at least 325

W/kg matched with at least 921 W-hr/kg was identified as a fuel cell – Li-Po solution for a defined 30 min mission resulting in reduced power system mass compared to using only Li-Po batteries. Utilizing enhanced specific power and energy fuel cell properties and current Li-Po battery properties, both a 30 min and 60 min mission were shown to have a reduced mass through hybridization. The feasibility and sizing strategy of hybrid-power components depends upon both mission requirements and power source component intrinsic properties.

Acknowledgments

As with any large undertaking, no one person is solely responsible for the results. In fact, a quote from Sir Isaac Newton is appropriate: “If I have seen further, it is by standing on the shoulders of giants.” This is indeed the case for me. If not for research performed by those who came before me, and support from AFIT faculty and my family, none of this would be possible. Perhaps an additional result not directly related to the topic is confirmation that we can only achieve what we first reach for. We usually can do it, if we only reach.

I thank Lt Col Harmon, Dr. Ryan Miller, Dr. Kelly Stewart, and Mr. J. Simmons for their assistance and guidance.

Paul M. Hrad

Table of Contents

Page

List of Figures

List of Tables

Page

Nomenclature

AFRL	Air Force Research Laboratory
AGL	Above Ground Level Auxiliary Power Unit
APU	Auxiliary Power Unit
AVL	Athena Vortex Lattice
BART	Basic Aerodynamics Research Tunnel
BATCAM	Battlefield Air Targeting Camera Autonomous Micro-Air Vehicle
BATMAV	Battlefield Air Targeting Micro Air Vehicle
BDAP	Boeing Design and Analysis Program (Software)
BOP	Balance of Plant
CA	Contributing Analysis
CAD	Computer Aided Design
CFD	Computational Fluid Dynamics
CG	Center of Gravity
COTS	Commercial off the Shelf
DAME	Distributed Analysis Modeling Environment (Software)
DARPA	Defense Advanced Research Projects Agency
DMFC	Direct Methanol Fuel Cell
DoD	Department of Defense
DOE	Design of Experiments
DYMORE	A Structural Analysis Code (FEA Software)
EADS	European Aeronautic Defense and Space Company
EW	Electronic Warfare
FAA	Federal Aviation Administration
FLOPS	Flight Optimization System (Software)
FEA	Finite Element Analysis
GenMAV	Generic Micro Aerial Vehicle
GPS	Global Positioning System
GUI	Graphical User Interface
HASC	High Alpha Stability Control (Software)

IED	Improvised Explosive Device
ISR	Intelligence, Surveillance, and Reconnaissance
MAV	Micro Aerial Vehicle
MDO	Multi-disciplinary Design Optimization
MEA	Membrane Electrode Assembly
MSL	Mean Sea Level
NACA	National Advisory Committee for Aeronautics
NEWSUMT-A	New Sequence of Unconstrained Minimization Techniques (Software)
NPSS	Numerical Propulsion System Simulator (Software)
NRL	Naval Research Laboratory
OEF	Operation Enduring Freedom
OIF	Operation Iraqi Freedom
OptdesX	Optimal Design for X Windows (Software)
PABLO	Potential flow around Airfoils with Boundary Layer (Software)
PEM	Proton Exchange Membrane or Polymer Electrolyte Membrane
PEMFC	PEM Fuel Cell
RAM	Rapid Aircraft Modeler
R/C	Radio Controlled
RMFC	Reformed Methanol Fuel Cell
SBIR	Small Business Innovation Research
SIGINT	Signals Intelligence
SOC	State Of Charge
SOFC	Solid Oxide Fuel Cell
SUAS	Small Unmanned Aircraft System
SUB3D	Optimization Software
UAS	Unmanned Aircraft System
UAV	Unmanned Aerial Vehicle
USAF	United States Air Force
V&V	Verification and Validation
VSP	Vehicle Sketch Pad

List of Symbols

a	acceleration
J	advance ratio
e	angle of thrust relative to flight path
ω	angular velocity
AR	aspect ratio
Φ	bank angle
$\overline{X}_{cg}$	center of gravity, normalized by chord length
Γ	circulation
θ	climb angle
C_D	coefficient of drag
C_{Di}	coefficient of drag, induced
C_{Dp}	coefficient of drag, parasitic
C_{Do}	coefficient of drag, zero lift
C_L	coefficient of lift
C_P	coefficient of power
C_T	coefficient of thrust
C_Q	coefficient of torque
C_Y	coefficient of side force
K	constant, drag polar
F	constant, Faraday
K_V	constant, motor
i	current density
I	current or second moment of area
I_0	current, no load
ρ	density, air
D	diameter
Δ	differential
q	dynamic pressure
η	efficiency
EDS_{SE}	energy dense source, specific energy
EDS_{SP}	energy dense source, specific power
$\Delta\overline{g_f}$	Gibbs free energy
g	gravity

L	lift
L'	lift/span loading
c_L	local lift coefficient
M	Mach number
m	mass
$\overline{X}_{np}$	neutral point, normalized by chord length
P	power
PDS_{SE}	power dense source, specific energy
PDS_{SP}	power dense source, specific power
R	radius
r_1	radius of curvature relative to vertical plane, local
R	radius, resistance, or universal gas constant
ROC	rate of climb
R_0	resistance, internal
r	resistance, specific
E	reversible electromotive force, voltage
RPM	revolutions per minute
Re	Reynolds number
n	rotational speed in revolutions per second or loading factor
e	span efficiency factor
T	thrust, or temperature
t	time
Q	torque, or charge capacity
V	velocity, or voltage
W	weight
c	wing chord, local
C_{ref}	wing chord, reference
W/S	wing loading
S	wing planform area
b	wing span

CONCEPTUAL DESIGN TOOL FOR FUEL-CELL POWERED MICRO AIR VEHICLES

I. Introduction

I.1. Motivation

I.1.1. Unmanned Aircraft Systems (UASs): For Today and Tomorrow

Execution of Operation Enduring Freedom (OEF) and Operation Iraqi Freedom (OIF) cost 5308 U.S. lives and wounded 36,350 U.S. personnel by the end of 2009.[1] The vast majority of these casualties occurred during OIF post combat operations when patrols and convoys became more susceptible to Improvised Explosive Devices (IEDs). Any additional reconnaissance and situational awareness critically aids the mission and those charged with accomplishing the mission. Because lives and resources are at stake, one of the most instrumental uses of UASs is Intelligence, Reconnaissance, and Surveillance (ISR).

However, UASs can be used for many additional mission essential applications. In addition to ISR, UASs could find useful employment in the placing of sensor equipment, relaying of communications, monitoring of the environment, controlling of crowds, surveying of land, and miscellaneous security.[2] They can be used for homeland security, law enforcement, border security, weather monitoring, and aerial video capture for sports events.[3] UASs can be tailored for "environmental compatibility"[4] with low acoustic, thermal, visual, or pollutant emission designs. In fact, UASs are ideal for any aerial job that may be dull, dirty, or dangerous.[5] The hand-launched and man-carried variety find applications to irregular warfare, such as personal ISR, Signals Intelligence

(SIGINT), cyber warfare, Electronic Warfare (EW), Counter-UAS, and automatic sentry duty.[6,7] These families of systems are quite limited currently, but with technological maturation will transform the way wars are fought.

The UAS market continues to grow. The Department of Defense (DoD) requested $5.4 billion in line items specifically designated to UASs,[8] and global UAS spending will easily reach $5.8 billion by 2014.[9] This high rate of expenditure indicates that UASs are becoming valuable commodities, and continued research and development is necessary for capability enhancement. Two beneficial paths of research and development include making the UAS smaller for ISR missions in and between buildings, and improving the energy capacity by including fuel cells.

I.1.2. Future UASs: Fuel Cell-Powered Micro Aerial Vehicles (MAVs)

The next step for technology is to offer all of the capabilities provided by UASs in a smaller package. Once sensor equipment became miniaturized to less than 18 g in the mid 1990s, the utility of MAVs became apparent.[10] Not only can MAVs be used for "over the hill" battlefield employment and "around the corner" urban reconnaissance, but also for non-typical applications such as hostage rescue and counter drug operations.[10] These lightweight aircraft can be carried by soldiers and launched by hand.[3] Some even see MAVs carrying small explosives.[11] The Air Force Research Laboratory (AFRL) strategic vision describes future MAVs as "ubiquitous sensors and shooters."[12]

MAVs can especially benefit from fuel cells because of stealth characteristics inherent to fuel cells. Low acoustic and heat signatures are even more important as reduced size allows the MAV to get closer to the target. Stealth applications and the potential for extended endurance make the fuel cell an important power source. Other

motivations for a fuel cell-powered UAS are for environmental research. Fuel cells can provide power with low pollution released into the atmosphere during use, and some types of fuel cells can do this without releasing any harmful pollution during use.[13] Observation or sampling tasks that require a closer field of view than afforded by satellites, such as vegetation mapping, deforestation or urbanization studies, and habitat analysis are all prime candidates for a low and slow remote sensing platform.

When comparing fuel cells with the use of rechargeable batteries, the Proton Exchange Membrane Fuel Cell (PEMFC) is only limited by the amount of hydrogen it can store, and can be recharged by replacing a hydrogen cartridge. This takes only seconds compared to hours that might be required for a battery recharge.[3] Power is continuous over the duration of fuel and oxidant availability.[13] A further cited benefit of fuel cell use is that individual fuel cell subcomponents may be distributed to some degree within the volume available. Currently, batteries exhibit excellent specific power, but often lack sufficient specific energy. Fuel cells, on the other hand, supply superior specific energy, but yield lower specific power. Furthermore, fuel cell efficiency and power-available can be significantly reduced when the fuel cell is subjected to a fluctuating load. As technology matures, the fuel cell will be the most attractive choice for low power, high endurance applications.[15] Because of the endurance fuel cells offer compared to high powered sources such as batteries or jet fuel, fuel cells can enable high altitude UASs to act as satellites at low relative cost compared to actually launching a satellite.[15]

Due to the future importance of MAV technology, a MAV steering group that includes representatives from AFRL directorates has established the strategic vision for

MAVs. Integrated Product Teams (IPTs) include propulsion, power, payload, CONOPS, and integration specialists. Because the future flavor of MAVs and their applications are so undefined, MAVs offer the perfect opportunity for conceptual design.[16]

I.1.3. The First Step: Conceptual Design of Fuel Cell-Powered MAVs

Reducing the size and changing the power source requires a system level analysis. Creating a fuel cell-powered UAS is much more than simply integrating a fuel cell into an existing battery powered UAS; due to low power to weight ratios and Balance of Plant (BOP) considerations, all other aircraft components should be designed around the fuel cell since it is the primary component.[17] A manufacturer of fuel cells, ProCore, says that when compared to batteries, fuel cells can achieve 62% weight savings, 50% size reductions, and 20% cost reductions.[3] The United States Air Force (USAF) needs an in-house tool to enable fuel cell technology to be integrated into MAVs. This conceptual design tool is a first step in the right direction.

This effort builds upon Lt. Mustafa Turan's work of establishing and demonstrating an integrated model of previously autonomous aircraft design software for the purpose of MAV design. He established a basis for further integration.[88] Where he focused primarily on aerodynamic issues of a tail-sitting coaxial MAV, this effort is geared toward the needs of the power systems engineer who must decide on an optimum power system. It could also lay the groundwork for future gains in MAV design.

Integration is complicated. Even though the internal combustion engine or jet engine design in itself is a complex multidisciplinary endeavor,[18] the design and subsequent integration within the airframe has matured to such a point that one might just think of the engine as merely another component. Economic and ecological assessments

now require the consideration of alternative and renewable power sources in any upcoming engineering endeavor, and the use of fuel cells appears to be just around the corner to widespread application. Unfortunately, very few fuel cell integrated conceptual design tools exist.[19] Not far off in the future, the design space of a fuel cell-powered aircraft, UAS or MAV, will be better known, and integration of a fuel cell will be as simple as replacing a battery, or even a jet engine. This conceptual design tool will be a foundation for future realization.

After a fuel cell integrated conceptual design tool for fixed wing MAVs is established, it is reasonable to branch out to a flapping wing or rotary wing MAV. A trade study between multiple configurations might be conducted, making general assumptions, and then revising the assumptions as this new body of research grows. With a tool such as this, if it turns out that a flapping wing configuration that requires a voltage converter uses less total power to do the job, it will be easy to identify. Or, if the theory predicts other modes of flight consuming less power, but technology is not ready, a study utilizing the conceptual design tool described in this thesis might set the groundwork for future gains.[16]

I.2. Background

The intent of the following paragraphs is to rationalize the need for a conceptual design tool for a fuel cell-powered MAV. First, the use of fuel cells in aircraft is discussed. Then, fuel cell use at the UAS and MAV size is discussed.

Hydrogen fuel cells have been considered for general aviation, but have so far been too impractical for widespread use. It is speculated that fuel cell efficiencies must

increase to 60% and power plant power to mass ratios increase to 600 W/kg for fuel cells and motors to compete with internal combustion engines run on hydrocarbon fuel.[20] Estimates vary, but fuel cell system specific power must increase 20-fold before all-electric vehicles will be used for general aviation. Though not technologically mature, Solid Oxide Fuel Cell (SOFC) power sources appear to be the most likely future candidate for internal combustion replacement. A comparison between fuel cell systems that can carry enough energy as an internal combustion engine showed that the SOFC can provide greater overall reduced aircraft weight and greater range than pure hydrogen based fuel cell.[13]

A National Aeronautics and Space Administration (NASA) study looked at the feasibility of using a compressed hydrogen PEM fuel cell in a propeller driven small electric light sport airplane. A commercially obtained Numerical Propulsion System Simulator (NPSS) code was used to simulate propeller thrust and calculate air/hydrogen flow rates. NASA's Flight Optimization System (FLOPS) code was used for aircraft mission performance. A supercharger was included in the fuel cell model in order to supply the required flow. Automotive fuel cell scaling allowed for the prediction of weight and volume. Due to power, weight, and heat management issues, this study determined that fuel cells could not yet compete with internal combustion engine aircraft, but it is feasible to produce aircraft with current fuel cell technology that can fly adequately with reduced performance.[14]

Often, when technology reaches a certain readiness level, it is simply inserted by direct replacement into old technology at the subsystem level. In many cases, direct replacement fails to exploit cross-functional benefits of the new technology, which

hinders total benefit at the system level. Instead of asking where a new technology can fit, one should ask which applications could benefit from the new technology. This has the advantage of leveraging the new technology at its fullest potential. One study applied this type of approach to fuel cell-powered aviation. After the concept space was defined, candidate architectures were determined. These capabilities were then mapped onto performance measures to identify a broader range of possibilities. Even when an expanded design space was considered to accommodate the fuel cell, the study determined that internal combustion engines are best suited for aviation today, especially where excess power is necessary. Due to fuel cell specific requirements and potential for dual-use application, the study also found that for long endurance missions not excess power dependent, fuel cells become the most efficient option.[21] This thesis will present high-level means to evaluate when fuel cells should be considered.

Fuel cell progress continues to intensify. Already, certain aircraft components such as an Auxiliary Power Unit (APU) have been replaced by fuel cells. This was done on the Airbus A320. Recently, the first propeller driven, 25 kW Antares DLR-H2 developed by the German aerospace center became the first manned aircraft to take off with all power coming from a fuel cell.[22] Fuel cell-powered aircraft should continue following the typical technology maturation path of reduced size and increased performance.

I.2.1. Propulsion Emphasis

With regard to aviation platforms, the reduced size of MAVs could drive consideration of additional means of propulsion. A considerable problem with development at the MAV scale is the lack of consensus on how a MAV should be

propelled. The shape of the vehicle could be based on a fixed wing platform, but could also be driven by a flapping wing configuration. The aerodynamic form may also be driven by a requirement that the MAV perch like a bird.[16] Each type has different power requirements and may have different optimal propulsion systems. Flapping wing MAVs make an attractive option due to biological similarity, but the required voltage may be quite high due to high power needs. Efficiency losses caused by on-board step-up voltage conversion make high power generation in small MAV packages very challenging.

A typical task for the electrochemical/thermal branch of AFRL might be optimizing the power for a given flight vehicle and mission, or determining what kind of mission can be supported with a given flight vehicle and power source. Some limited power profiles are available for some platforms, but experimental conditions of the sparse data are often not available in enough detail to be useful. However, since such a standard MAV flight vehicle has not been yet defined, and since power options such as fuel cells are more intertwined with the airframe, both questions could benefit from a conceptual design tool for MAVs.

I.2.2. MAV Size

The literature exhibits a wide range of sizes for MAVs. Perhaps the most prevalent dimensions are those initiated by a Defense Advanced Research Projects Agency (DARPA) workshop in 1995 that precluded a phase one Small Business Innovation Research (SBIR), indicating that a 15.24 cm MAV was feasible.[10,23] As such, air vehicles with no linear dimension greater than 15.24 cm became the most well known standard for the maximum dimension of a MAV. Further refinement of MAV size

incorporates a weight of roughly 200 g.[24] Currently, the technology exists to build a MAV with Commercial off the Shelf (COTS) subcomponents weighing only 14 g with a 9.9 cm wingspan.[25] However, not everyone accepts this convention. Reference 26 generally agrees with the maximum length to be less than 20 cm, but allows the mass to be up to 500 g. Increasing in size, reference 27 allows a wingspan to be between 15.24 cm and 30.48 cm with a chord length of about 10.16 cm. A solar powered "MAV" called the Sunbeam I used a 38.1 cm wingspan.[28] At the lower size bounds, truly small flight vehicles could also be called MAVs. Cutting-edge research continues into insect sized MAVs that are usually referred to as *nano* aerial vehicles, but technology readiness is still far off.[12]

The previously mentioned MAV Steering Group has suggested that even though the long-term goal for MAV size is that of an insect, a reasonable size goal for UAS miniaturization is that of a Cooper's Hawk.[16] Currently, this is the most realistic size needed to exploit fuel cell technology. The average male Cooper's Hawk ("accipiter cooperi") is 0.312 kg with a wingspan of 0.71 m, and the average female is 0.500 kg, with a wingspan of 0.83 m.[30] It flies at 11.2 m/s with a planform area of 0.09 m^2.[31] For comparison, other birds in this class are the broad-winged hawk ("buteo platypterus," at m = 0.385 kg, b = 0.84 m, S = 0.101 m^2, and V =9.9 m/s), and the American coot ("fulica americana" at m = 0.443 kg b = 0.64, S = 0.06 m^2, and V =13.8 m/s).[31]

Although not common nomenclature, a class of aerial vehicles sized between UASs and MAVs exists within this undeveloped architecture. Mini-UASs are a representative class of UASs defined as those with a maximum wingspan of 70 cm.[32] Examples of MAVs within this size envelope that are in service are the Battlefield Air

Targeting Micro Air Vehicle (BATMAV), which is Aeroenvironment's WASP III, the Battlefield Air Targeting Camera Autonomous Micro-Air Vehicle (BATCAM), which is an AFRL research platform, and the Generic Micro Aerial Vehicle (GenMAV), which is one of AFRL's newest research platforms. Notably, the USAF UAS Flight Plan from 2009 classifies these aircraft as Small Unmanned Aircraft Systems (SUASs).[6]

For a starting point, by 2015, the MAV size goal is that of a Cooper's Hawk with mass of 0.350 to 0.500 kg. By 2030, insect -sized MAVs should be attainable. An initial concept would be to send out a MAV for 30 days, and allow it to perform daily one-hour missions while recharging by some means between sorties. The MAV would be "biometrically compatible," either visually or mechanically, so that it could hide in plain sight.[16]

The following information is included in order to share an appreciation for the sizes, along with some tradeoffs that can be expected in both natural flight and aerospace engineering. Figure 1 illustrates the geometric similarity between speed and weight across species from insect to bird, and then all the way up to a Boeing 747.[31] The trend line equates wing loading to a cubed constant multiplied by the square root of weight. Entries to the left of the trend line show potentially underpowered insects, birds, or aircraft. Entries to the right show highly maneuverable insects, birds, or aircraft. This figure provides a snapshot of reasonable characteristics one would expect to find in a new flying vehicle. Not only does it provide quantitative values for wing loading, but also gives insight into the realm of the possible.

As mass increases, velocity must increase to support the wing loading. Likewise, as mass decreases, the velocity is expected to drop as well. The problem with reduced

velocity is that the aircraft has less flying capability in environments with even modest disturbances. At best, the aircraft might not get to its destination. At worst, it could be blown out of the sky. Figure 1 demonstrates a strong physical relationship despite size, but the disparity between these propulsion and fuel mass fraction numbers is empirical evidence that propulsion requirements are not that clear. Decreased propulsion and fuel mass fraction should be the main goal for MAV research. MAVs must fly faster to support additional weight required by inefficient propulsion, despite possible operational requirements more suitable to slower flight.

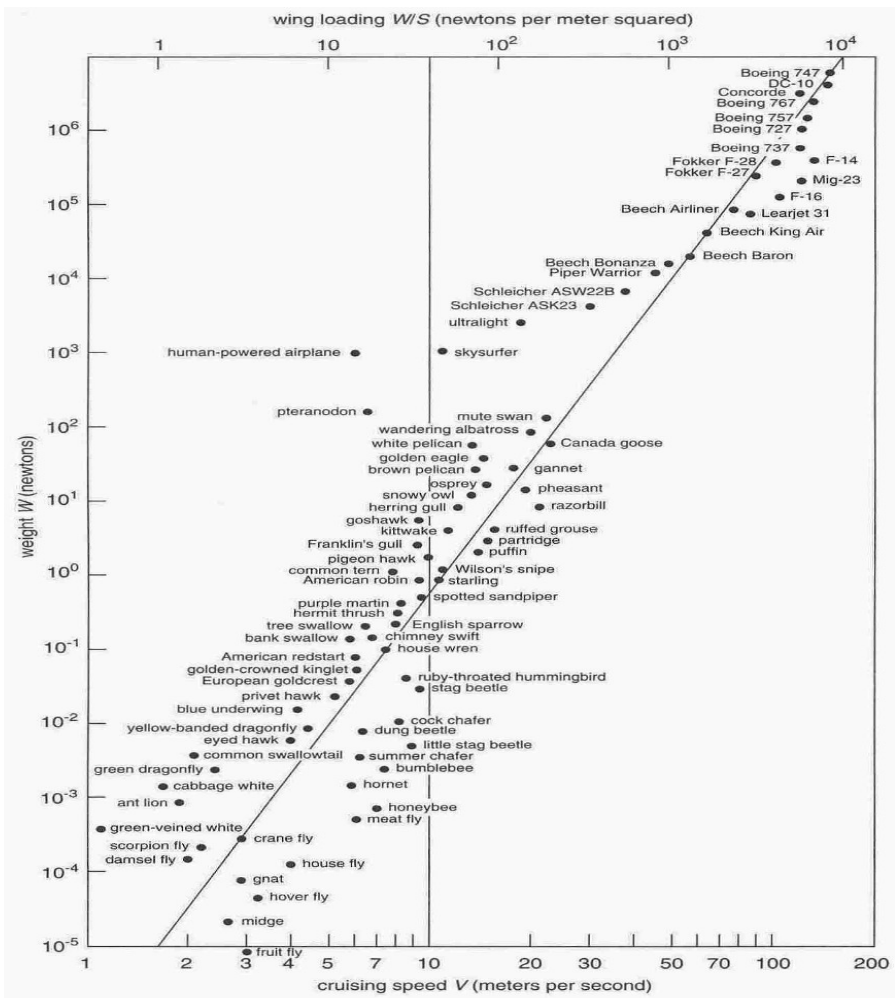

Figure 1. **Wing loading, weight, and speed of small insects up to jumbo jets.**

For this study, power-required better expresses the metric typically conferred by size. See Table 1 for general correlation.[3] The power boundaries are not yet rigid at this point in technological development.

Table 1. Power requirements for fuel cell-powered air vehicles

	Low [W]	High [W]
Large UAS	10^5	10^5+
Medium UAS	10^3	10^4
Small UAS	50	300
MAV	1	20

Although Figure 1 shows that a smaller platform should fly at slower speeds, to overcome the propulsion inefficiencies, higher velocity and more power is required.[56] Here lies a challenge for MAVs. It is very difficult to package enough power and energy for maneuverability and endurance within the MAV size. The tradeoffs require an optimization: either carry just enough fuel (low power) and risk the perils of slow flight, or carry much more fuel (higher power) to fly faster. Better performing MAVs of the future will likely have increased power than those values shown in Table 1, but this is limited by technology maturation rates of available power sources.

I.2.3. Modes of Flight

Modes of flight typically fall within three varieties: fixed wing, rotary wing, and flapping wing. Each mode has its merits and cost. With the intent of designing MAVs with the lowest possible power required, it is sensible to consider which mode offers flight with minimum power. A study compared theoretical calculations, performed a literature review of small bird and insect flight, and carried out experimental analysis to

find power requirements for fixed wing, rotary wing, and flapping wing flight. It was determined that for the 50 g class subject to low Reynolds number flight, fixed wings required the least power and offer best efficiency when no hover requirement exists. When there is a hover requirement, flapping wings offer best efficiency at low speed flight. When both high speed and hovering is required, rotary wings are most efficient.[33] Unfortunately, flapping wing flight can require four times the power when compared to fixed wing.[31]

I.2.4. Unique Issues with Respect to MAVs

The Wright Brothers and others fundamentally understood the application of flight technology by the first decade of the 20th century. Through experiment and dogged perseverance, these innovators devised the concepts that have held for over 100 years. Once the potential of UASs took root, technologists and engineers tried to apply the same lessons learned from full-scale aircraft to smaller platforms. In the case of UASs, however, individual subcomponent efficiencies had to improve in order to make useful systems. Most critically, the propulsion system had to get a lot more efficient.[10] The reason for this is that aerodynamic control becomes more difficult at lower moments of inertia and smaller size. Operation within gusts and crosswinds further complicates the problem. Avoidance of obstacles and ground/wall effects are significant issues.[34]

Aerodynamic similarity separates MAVs and smaller vehicles from UASs and larger vehicles. The most important differences come from flight in the low Reynolds number regime and the use of low aspect ratio planforms. Because of these aerodynamic phenomenon, the miniaturization of aircraft to MAVs hit a technological speed bump after reaching the small UAS (SUAS) class.

I.3. Research Objectives

The propulsion directorate (AFRL/RZ) is fielding a SBIR regarding flapping wing MAV feasibility. Current power options are limited to batteries, fuel cells, and capacitors while a future option might include biological power. Near-term, analysis of hybrid systems made of electrochemical/thermo combustion is being done to achieve required energy and power.[16] Though this research does not directly relate to the issue of flapping wing aircraft, the conceptual design tool will form the basic structure in which a flapping wing system could be integrated.

This research will build upon the idea that various design software packages can be integrated together into a useful, multidisciplinary tool for MAV design. The intent is to tailor the product for the Power and Propulsion branch of AFRL so that they may evaluate various power schemes used for MAVs. This could assist in directing the application of resources and may point to solutions that otherwise might not have been realized.

This tool must be able to calculate the first-order power draw from the propulsion system over time for a given airframe and mission. A power profile must be generated showing current and voltage for take-off, climb, cruise, loiter and other mission segments.

After simulating the average segment power of each mission segment, the power system mass will be calculated using specific energy and specific power values for a hybrid system. The hybrid system will be made up of a high specific energy component, which could represent a fuel cell, and a high specific power component, which probably would represent a battery, but could represent a capacitor.

I.4. Hypothesis and Scope

A useful conceptual design tool can be built to enhance the ability of the propulsion/power specialist to analyze and compare MAVs and MAV propulsion/power strategies. The objective of this research is to focus on the propulsion power requirements of MAV design, while maintaining adequate system-level analysis to improve the mission-dependent selection and optimization of the power source.

In this proof-of-concept endeavor, certain disciplines have largely been ignored and can be included later. For example, structural integrity, aeroelasticity, material selection, and any volume related requirements are not modeled. Active control strategies and navigation are also not incorporated at this initial stage. Furthermore, this tool only addresses fixed wing configurations and does not model flapping or flexible wings.

Efforts of this thesis are initiated within the mini-UAS class, which is at the larger end of the MAV class. Future development is geared toward smaller MAVs. For reasons explained later in the text, the tool developed here will also be applicable for larger classes of air vehicles such as UAVs that fly at a chord Reynolds numbers of approximately 10^5 or greater.

In response to the sponsor's need for power system optimization, this tool analyzes a power system based on a set airframe, although iterations on MAV design could easily be performed with only minor changes. A much greater need exists for near-term integration opportunities than farther off feasibility studies, so this tool is tailored to provide realistic information and answer questions about what can be done today.

Much discussion is provided within this thesis regarding the importance of building the aircraft design around a well-integrated fuel cell and Balance of Plant (BOP). This remains the end goal. Any complex effort can be simplified into a series of steps. The first step toward the goal is to develop a power system analysis tool that incorporates system level aerodynamics. With a mission power profile, combinations of high specific energy and high specific power components can be evaluated and used in the most appropriate way.

I.5. Methodology

This section lists the key steps performed under this research. A considerable portion of this effort was finding, integrating, verifying, and validating the software within the Model Center framework. Before anything else could be done, the framework had to be built. This involved integrating suitable programs such as QPROP and AVL into Model Center. Since QPROP and AVL require a specific airframe, an initial task was to find an applicable MAV airframe and build an adequate model. Some possible alternatives were the GenMAV, the BATCAM, or the BATMAV. Building the AVL and QPROP model required considerable effort, and included some speculation as will be discussed. This was due to the unavailability to certain information. Integration within Model Center required an in-depth understanding of how each subcomponent operates.

Once the aerodynamics could be established with AVL and QPROP, each component needed to be thoroughly verified for proper execution within the Model Center environment. This involved many iterations and continual evaluation of results. Development of a means to estimate a theoretical power requirement necessitated

incorporation of additional tools such as Mathwork's Matlab and Microsoft's Excel into the Model Center environment. Development of how to choreograph all subcomponents to run a designated mission proved challenging. Once the mission driver component was built, the power profile became available for analysis.

With the power profile in hand, various means of satisfying the mission dependent power needs could be studied. The tool assumes that a high specific energy source and a high specific power source will be used together to minimize the overall power system mass.

All of these results were verified as well as possible by comparing to available literature. This study seeks to assess near-term feasibility of using a fuel cell and to identify trends and options for a hybrid-power system with trade studies, parameterization, and optimization.

I.6. Thesis Overview

Chapter 1 discussed the importance of MAV research and why a conceptual design tool is necessary. Clearly, MAVs are instrumental in the wars of today and enhanced capability could dramatically change the way in which wars are executed tomorrow. A brief background explained that this effort is being initiated in order to improve the propulsion subsystem, which has been identified as a key MAV research area. Some peculiarities due to small MAV size, such as low aspect ratio wings flying in low Reynolds number regime, and how they are related were discussed. Lastly, a hypothesis was offered explaining that a fuel cell-powered MAV conceptual design tool

can aid the power system designer. The steps and goals of the research were then explained.

Chapter 2 includes a more detailed literature review of some fielded MAVs, important UASs and MAVs that integrated fuel cells, and key software used for design. The transition between what has been done and what is being done is made in Chapter 3. Chapter 3 addresses the specific steps taken to develop a conceptual design tool within Model Center. Results and analysis are provided within Chapter 4. Finally, conclusions, lessons learned, shortfalls, and recommendations are made in Chapter 5.

II. Literature Review

Since aircraft conceptual design is a multidisciplinary endeavor, many components and subcomponents must work together to build a useful product. Each discipline offers important challenges with respect to fuel cell-powered MAV conceptual design, and a discussion of all issues would quickly grow beyond the scope of this effort. This chapter is designed to provide a broad view of some types of MAVs that benefitted from conceptual design, followed by critical subcomponents such as aerodynamic software, fuel cell concepts, and propulsion software. The chapter ends by integrating all subcomponents in the Model Interactions section. Equations are provided where they may aid in understanding, but the details of the actual models used will be presented in Chapter 3.

II.1. Notable System Level Efforts

II.1.1. Technology Survey

Over the last 15 years, many MAV attempts have been made, but most were not well known. A technology survey[2, 24] found in the literature highlights some successful MAVs. A short list includes the DARPA funded Black Widow and WASP, both produced by AeroVironment, as well as a number of commercial crafts: Mite by US Naval Research Laboratory (NRL), MicroSTAR by BAE Systems, DO-MAV by European Aeronautic Defense and Space Company (EADS), and Trochoid by MLB.[10] These crafts tend to use a single, conventional, center-line tractor propeller design with carbon fiber or structural foam bodies. DARPA and AeroVironment's Black Widow

were catalysts for widespread MAV research. In order to design a craft capable of carrying the required surveillance equipment, a Multi-disciplinary Design Optimization (MDO) methodology was used.[23] Currently, numerous companies and agencies are considering a wide variety of MAV options. One such non-conventional MAV is the Sunbeam I: A solar powered MAV weighing 71 g, using 3.5 W, with a span less than 38.1 cm.[28] MAV technology continues to mature as researchers, students, and hobbyists participate in the annual International Micro Air Vehicle Competition that started in 1997. The winner must use the smallest MAV to observe a target located at some distance, and transmit the information back to the judges.[10]

II.1.2. MAV Successes

In addition to the MAVs listed in the *Technology Survey* section, MAVs discussed here are those in which the conceptual design techniques have been more readily accessible in the literature. The most successful and well-known MAV is arguably the Black Widow. MDO consisted of five components: vehicle and propeller aerodynamics, battery and motor performance, and a weight buildup. Vehicle aerodynamics used lifting line theory for induced drag, Blasius skin friction formulas for friction drag, Hoerner equations for interference drag, and equivalent parasite areas for produrbance drag. A minimum induced loss methodology determined the propeller aerodynamics. The motor and battery performance relied on experimental data. A design envelope was defined and an Excel solver determined the optimal configuration that enabled the 80 g MAV to fly up to 234 m high, 1.8 km out, for 30 minutes. A notable result of this effort is that on the MAV scale, 80% propeller efficiency is possible.[23]

After the Black Widow showcased MAV possibilities, students and universities drove research and MAV design. An extraordinary effort came from Brigham Young University. They utilized straightforward aircraft design equations and techniques to perform a feasibility study on a solar powered MAV called the Sunbeam I.[28] The following equations drove the design.[36] By setting the lift, L, equal to the weight, W, the coefficient of lift, C_L, for steady, level flight can be determined with Eq. (1),

$$L = W = \frac{1}{2}\rho V^2 S C_L \qquad \textbf{(1)}$$

when the density, ρ, velocity, V, and planform area, S, are known. Power-required from the power source, P_{req}, can be determined by multiplying the thrust-required, T_{req}, by the propeller by the freestream velocity, V, and dividing by the propeller efficiency, η_p, and motor efficiency, η_m, as in Eq. (2).

$$P_{req} = \frac{T_{req} V}{\eta_p \eta_m} \qquad \textbf{(2)}$$

Thrust-required by the airframe, T_{req}, is determined by setting the thrust equal to the drag, defined by the drag coefficient, C_D, in Eq. (3).

$$T_{req} = \frac{1}{2}\rho V^2 S C_D \qquad \textbf{(3)}$$

The drag polar is defined by Eq. (4), which includes the zero lift drag coefficient, $C_{D,0}$, and K defined by Eq. (5).

$$C_D = \left(C_{D,0} + K C_L^{\ 2}\right) \qquad \textbf{(4)}$$

$$K = \frac{1}{\pi e_0 AR} \qquad \textbf{(5)}$$

where AR is the aspect ratio and e_0 is the span efficiency factor. These same equations were used within the research presented in this thesis.

Soon, MAV interest became high enough for the International Society of Structural and Multidisciplinary Optimization to hold a MAV competition in 1998. One team used a multidisciplinary design optimization software called NEWSUMT-A to create a potentially nonconventional aircraft for remotely piloted reconnaissance. Wing loading was set to 53.9 Pa and a flight Reynolds number goal was determined to be 2.0×10^5. A Selig airfoil was selected based on its ability to fly at expected stall Reynolds number of 10^5. A vortex lattice model was used to compare various planforms by minimizing induced drag, and empirical skin friction coefficients were integrated to estimate viscous drag. This analysis showed that a biplane would be the best choice. Geometry size and control-surface fractions came from the MDO. The vortex lattice program SUB3D was linked to the MDO for aerodynamics.[29]

MAV design became so popular that the University of Florida began an annual competition to promote MAV research. Each year, the winning MAV became smaller, while its utility became greater. Students and researchers often submitted entries as part of class projects. In 2000, Notre Dame designed a MAV for the 4th annual Micro Aerial Vehicle Student Competition.[37] This MAV used a small internal combustion engine for propulsion and a nickel-cadmium battery for payload power. Today, they might have chosen to use an electric motor and a lithium ion battery instead due to technology maturation. Also, at the time, little was known about low aspect ratio and low Reynolds number flight, so experimental tests were performed on multiple planforms to determine optimal MAV design. During conceptual design, it was desired to select the planform

that offered the lowest angle of attack at the lowest drag coefficient. Using a force balance and the speed required for chord Reynolds numbers between 7.0×10^4 and 1.4×10^5, it was determined that the Inverse Zimmerman planform was best. When viewed from above, this planform has a circular leading edge and an elliptic trailing edge. A second-degree polynomial curve fit proved that the best aspect ratio was 1.5 flying at an angle of attack of 10°. Brigham Young University participated in the sixth International Micro Air Vehicle competition and created a 14 g, 9.9 cm span, 2 min endurance MAV by utilizing the OptdesX optimization software.[25] Known airfoils were analyzed with Java-Foil, and new airfoils with X-foil. The resulting airfoil required a positive pitching moment and a good lift to drag ratio at Reynolds numbers between $7.0 \times 10^4 - 9.0 \times 10^4$. Additional research determined that a 6% camber and 3% to 4% thickness is best when Re is less than 10^5. The reflexed trailing edge of an E330 flying wing airfoil was combined with an S7055 low Reynolds number airfoil leading edge. JavaProp and wind tunnel testing determined the optimum propeller, and COTS components were picked for the battery and motor that best reduced an optimization penalty.

Air Force Special Operations currently flies AeroVironment's WASP III under the BATMAV (Battlefield Air Targeting Micro Air Vehicle) program. This 72.3 cm wingspan MAV uses rechargeable lithium ion batteries and an electric motor. It has provided beyond-line-of-site capability to Airmen since 2007.[38] The BATMAV batteries can be charged in the field with the Jenny-BRITES fuel cell charging system.

Obviously, MAV interest is increasing and more research is necessary for better capabilities. One vehicle called the BATCAM became surplus inventory when it was not

chosen as an operational asset, yet it has been very useful for research collaboration between AFIT and the AFRL. This reliable test platform has been invaluable to navigation, aerodynamics, and controls research. It has a 53 cm wingspan and uses lithium polymer batteries. It could easily become a fuel cell test craft as well.[12] However, testing revealed some deficiencies that were better addressed through a new platform. The AFRL Munitions Directorate recognized this opportunity and developed a research MAV called the GenMAV that it hopes will be adopted as a standard MAV research platform. The GenMAV has a wingspan of 60 cm, planform area of 0.75 m^2, and is designed for a cruise speed of 13.4 m/s. It was designed to fit within the mini-UAS or larger MAV class because this is the size best positioned for useful research near-term gains.[39]

II.1.3. Hybrid-Powered Fuel Cell UASs/MAVs

Despite an increase in complexity, power source hybridization has the potential to increase endurance.[40] Between the use of various types of batteries, fuel cells, capacitors, or internal combustion engines, matching parameters of the power source to the power sink appears to be key in overall power minimization. In many cases, current systems will be retrofitted with a combination of smaller batteries and fuel cells. The following hybrid-power UASs and MAVs have been built for either demonstration or actual military service.

AeroVironment's WASP, one of the original successful DARPA MAVs, became somewhat of a test bed for the use of fuel cells within MAVs through the creation of the Hornet. This slightly heavier, 380 g MAV integrated a 10 W PEMFC into the wing surface for a dual purpose (structural and power) application. It used sodium borohydride

hydrogen storage. In today's multidisciplinary design environment, any double application can pay dividends.[41] This platform demonstrated that a fuel cell-powered MAV could be built, but as a demonstrator, the Hornet did not add operational capability not already available with batteries.

In order to break the UAS endurance record, a 150 W compressed hydrogen PEM fuel cell and 2100 mAh lithium polymer battery hybrid-power system has been produced to fly a craft weighing 5 kg with a 4 m wingspan. This was California and Oklahoma State University's bid for the Federation Aeronautique Internationale competition within the R/C class powered by "any single or combined source of current." Fuel cells are obvious candidates for endurance power due to high specific energy and steady power production. In this case, 60 W were needed for optimal cruise efficiency and an additional 75 W were provided by the battery for take off. This effort illustrates some of the unique integration concerns when using a fuel cell. First, fuel cells require a specific amount of humidity for proper operation. This fuel cell was self-hydrating, but needed ambient air for the reaction and cooling. In order to provide air, inlet and exhaust ducts had to be incorporated into the structure. Power hybridization was regulated with PMOSFET gates and a Schottky diode was used to protect the fuel cell from incoming current. These losses due to hybridization were calculated as 0.2% of total energy available.[42]

At the large end of the spectrum, the Georgia Institute of Technology designed and built a 6.7 m wingspan, 500 W PEM compressed hydrogen fuel cell-powered UAS that they believed to be the largest fuel cell power plant used for aircraft yet. The

technology is not ready for general aviation, but the hope is that efforts such as these will motivate further research toward full-scale fuel cell passenger aircraft.[43]

After Georgia Tech's UAS success, the Office of Naval Research created the Ion Tiger, which is a 500 W PEM compressed hydrogen fuel cell driven UAS that will carry a 2.27 kg payload for 24 hours. To facilitate a rapid acquisition schedule, they leveraged technology from the automotive industry instead of tailoring the fuel cell design for the mission. Aviation specific fuel cell research will likely provide even longer endurance on future platforms.[44] The Naval Research Laboratory also created the Spider-Lion MAV. This 1.8 kg craft used a 100 W PEMFC with storage capability of 15 g of hydrogen gas.

Though a bit larger than the Ion Tiger, AeroVironment's 2.6 m wingspan small UAS called the PUMA proves that hybridization can work. A Protonex fuel cell said to perform at a peak 1000 W/kg and a lithium ion battery can fuel the hand-launched 5.7 kg PUMA for 9 hours. SBIR cooperation between AFRL and AeroVironment continues to push the performance envelope.

II.2. Documented Conceptual Design Efforts

Design and development of aircraft is arguably one of the more difficult engineering challenges of the 21st century. Many disciplines must come together and collaborate to meet system level objectives. Aerodynamics, thermodynamics, propulsion, structures, and stability/control are some areas critical for satisfactory design. Unfortunately, the optimization of one specialty area usually hinders other specialty areas. Because of these complexities, production aircraft are designed through iteration among the various design groups.[45] To make things more challenging, technology is

always in a state of flux. Conceptual design occurs at the “big picture” level. At this point, first-order relationships and models provide the designer a feasibility analysis, with detailed design analysis postponed. The speed of this analysis usually comes at the cost of its fidelity.

Relatively few conceptual design studies have been accomplished specifically for fuel cell-powered UASs and MAVs, but the Georgia Institute of Technology has done extensive work in this area.[15, 19, 46, 47, 49, 50] The common theme among all of their work is to assemble a set of models cast as Contributing Analysis (CA), and let an optimization routine determine the design variable values within a designated design space. The CAs are often parameterized low-fidelity approximations.[50] It is desired to let the design space be as large as reasonably possible. Much of their work, however, focuses on a technology demonstrator the size of a medium UAS. Despite the size difference from MAVs, much can be learned from their efforts. The following paragraphs provide relevant information on their research.

II.2.1. Variable Fidelity

Georgia Institute of Technology has developed a conceptual design environment that provides the designer the ability to select the depth of analysis required at the conceptual design stage. The depth of analysis is proportional to the computational cost and the accuracy of results. Obviously, when the design architecture is completely unknown, at the beginning, a large number of quick trade studies provide invaluable information on the general form. Later in the design, when the form and overall specifications are better known, a more thorough and complex analysis is desired. When a large design space is available for analysis, one might want to use many models of

varying fidelity to yield solutions not apparent at the start. For this purpose, Georgia Institute of Technology utilized Model Center to integrate an assortment of subtools.[46] The model choices for airfoil analysis included an inviscid airfoil program called Potential flow around Airfoils with Boundary Layer coupled One-way (PABLO) for low-fidelity and XFOIL for medium-fidelity. Aircraft geometry was modeled in 2-D for the simple model, and NASA's Vehicle Sketch Pad (VSP) was used for medium-fidelity. These geometry tools were used as a basis for friction drag calculations by way of a low-fidelity empirical form-factor method and high-fidelity airfoil numerical integration. Structure was represented through regressed equations for low-fidelity and a finite element program called DYMORE for medium-fidelity. Lastly, lookup tables were used for low-fidelity propulsion and an assortment of models incorporated into MATLAB/Simulink represented high-fidelity propulsion. Power management for hybrid architectures relied on a surrogate model for specific energies. Sizing is dependent on a single point in the flight envelope, the mission, and cruise conditions. Because the overall design model is verified by simulation using the high aspect ratio Pathfinder Plus, only combinations of solar and compressed hydrogen/oxygen are considered for nonconventional propulsion.[46] Other researchers on this team took the work further by identifying a method of reducing uncertainty that included a sensitivity analysis meshed into the optimization technique.[50]

II.2.2. Fuel Cell Case Studies and Limiting Assumptions

A closer look at fuel cell conceptual design examines some assumptions used to create multidisciplinary fuel cell UAS designs.[15, 47] Convenient assumptions are of two main varieties. The first is that a simple and direct replacement of a battery with a fuel

cell into an aircraft will produce desired aircraft performance. The second assumption is that fuel cells designed for other purposes, such as cell phones or automobiles, will scale to produce the desired aircraft performance. Both assumptions greatly simplify the design process, but often fail to incorporate enough subsystem detail required for overall performance optimization. The study that identified a lack of fuel cell specific analysis offers a model for hydrogen storage tank thickness, hydrogen storage tank mass, and fuel cell efficiency. Wings2004, a potential flow analysis code, was used for airframe analysis, Goldstein's vortex theory of screw propellers was used for propeller performance,[51] and propeller-fuselage interactions were calculated from Lowry's *Performance of Light Aircraft*. Results showed that when the BOP and the fuel cell stack are decoupled, and then compared to a conventionally sized fuel cell using power system specific energy and energy density as performance proxies, the optimization routine called for a larger fuel cell stack and a smaller BOP. In other words, treating the BOP as an individual component within the airframe not directly linked to the fuel cell provided a better system-level result. This result indicates that a fuel cell should have a larger stack to run at a lower current, closer to its most efficient operating point. Operating there will allow for greater endurance time when compared to the conventionally designed propulsion system. Including fuel cell and aircraft interactions, albeit with a more complex model, leads to a much bigger fuel cell for the same weight, offering greater endurance. For endurance, the fuel cell BOP should be adequate for 45-50% of active area limited power, and not 90% as used in the automotive industry.[47, 48, 52]

The Georgia Institute of Technology took a closer look at the feasible design space for a PEM fuel cell-powered medium UAS and built a matrix of alternatives that

included thousands of options.[15] Prior experiences drove most of the down-selecting, but the rest was left to an optimization technique that included areas of aerodynamics, propellers, propeller interference, fuel cell performance and weight, fuel cell BOP, compressed hydrogen storage, electric motor, wing weight, and performance. A low Reynolds number airfoil was selected. Then, Wings2004 provided induced drag and lift. Parasitic drag was calculated from experimental data. Propeller analysis was based on Goldstein's vortex theory of screw propellers using the Betz condition.[51] Fuselage and propeller interaction effects are based on the Lowry method. Static polarization curves and published data for Gore 58 series membrane electrode assemblies were used to model fuel cell performance, with max current density of 1100mA/cm^2/cell and max specific power of 0.6 W/cm^2/cell. The BOP power consumption of the compressor, water pump, and radiator was found by scaling. The motor was modeled as a lumped parameter equivalent circuit and overall performance was calculated using drag polar and gross weight. Cruise performance was found by matching required thrust at pre-stall airspeed and climb performance is based on maximum available thrust. The most important finding related to specific energy. If specific energy is the electrical output at cruise divided by aggregate system weight, it was confirmed that maximizing specific energy does not provide maximum endurance or maximum range. The hydrogen and hydrogen storage amounted to 48%. The fuel cell and BOP was 21%. When including the fuel cell, BOP, and hydrogen storage, this long endurance, medium sized UAS had a total propulsion mass fraction of 69%.[15]

More experimental in nature, another effort at the Georgia Institute of Technology utilized a design methodology for the conceptual design of a PEMFC UAS.[49] Despite

their objective size being larger than the MAV class, their work illustrates an approach to multidisciplinary design. The design space was limited to six fuel cells with varying capacity storage (18-370 standard liters), several propellers of varying length and pitch, aspect ratio of between 10 to 20, a designated wing area range, and a standard tractor propeller and empennage. It had to be within the R/C aircraft class under 25 kg. Aerodynamic analysis was performed with a Rapid Aircraft Modeler (RAM) to generate geometry inputted into the Boeing Design and Analysis code (BDAP), which was used to calculate skin friction drag. VORLAX vortex lattice code was used to find lift and induced drag and Goldstein's Vortex Theory of Screw Propellers was used as the propeller model.[51] Motor analysis was done with a lumped parameter model, with fuel cell analysis based on weight, power, and volume. Contrary to use in an automobile, since a fuel cell for aircraft generally operates in near steady state, polarization curves can be used to determine performance.[52] Polarization curves are plots of voltage verses area normalized current. Over 15 million combinations resulting from the possible motors, gear ratios, fuel cells, propellers, storage tanks, aspect ratios, wing areas, and aerodynamic configurations resulted in 2000 feasible cases. It is notable that almost no combinations were feasible under 300 W. The most important aspect of efficiency was matching the motor and propeller with the fuel cell's peak operating current at cruise.

It was shown that when the design task starts conceptually, and then later becomes more detailed, large areas of the design space could be cleared and more focused, thorough, and computationally intensive analysis could be done within a more narrow design space. Furthermore, this study identified that fuel cell BOP should be considered at the conceptual design stage.[49]

In conjunction with the above effort, they also sought to better characterize the fuel cell-powered aircraft design process by breaking up the fuel cell power plant system into segments and integrating them into an aircraft model.[19] The power plant consists of the fuel cell stack, temperature control system, air management, and hydrogen storage management system. A 500 W, 32 cell PEM self-humidified hydrogen-air fuel cell was selected, and the static polarization curve provided performance data. Various pumps and compressors must be used to regulate fuel flow, air flow, and temperature. Additional equipment is necessary to create the optimal conditions for 90% hydrogen utilization. Just as efficiency was considered for the fuel cell power supply, efficiency was also important in propeller selection. Efficient propulsion comes from large propellers, driven slowly by high torque motors. This demonstrator/proof-of-concept fuel cell aircraft attained a propulsion system that was 57% of the total weight. This relatively high value is partially attributed to a big downscaling challenge: the compressed hydrogen storage tank. As it becomes smaller, its weight fraction increases due to the structural needs of containing high pressure gas. Flight tests revealed that energy efficiency, taken by comparing propulsive hydrogen high-heating-value fuel to rotational energy, was near zero at idle, 14% during full power climb, and 18% during cruise. A full 15% of electrical output power was required by the BOP to sustain cruise.[19]

II.3. Aerodynamic/Stability Codes

MAVs offer unique aerodynamic challenges such as flight within low Reynolds number regime, low aspect ratio wings, thin airfoils, and limited experimental data. These challenges cannot be met with standard aerodynamic software and require

something specifically tailored to the flight regime. High accuracy in aircraft aerodynamic design may be achieved by numerically solving the full three-dimensional Navier-Stokes equations. This comes at great computational cost and time. Because of these costs, inviscid solvers are often used to speed up the process. Though acceptable for UASs and larger aircraft, this simplification is known to yield results that do not adequately predict performance of small MAVs due to the high relative influence of wing tip vortices.[53] This section begins with an explanation of why low Reynolds numbers and low aspect ratios require special attention, and then introduces several useful aerodynamic/stability software codes applicable to MAV design.

II.3.1. Low Reynolds Numbers and Aspect Ratios

Unlike an average UAS built by downscaling a general aviation aircraft, MAVs encounter both the challenge of subcomponent miniaturization and less-than-fully-understood aerodynamics. Given these challenges, one might compare MAV development with the Wright Brother's effort, when fundamental relationships were identified empirically.[24] Of all aerodynamic measures, the Reynolds number is most important. The Reynolds number quantifies the ratio of inertial forces to viscous forces. Large airplanes, such as those used for general aviation, fly in the high Reynolds number regime. For comparison, a small Cessna aircraft flies within the Reynolds number regime of 10^6.[11] Naturally, high mass and high velocity permits inertial forces to dominate. Since MAVs have low mass, inertial forces are low when compared to strong viscous forces within the flight regime, leading to low Reynolds numbers on the order of 10^4 to 10^5.[26, 27, 54, 55] Conventional fixed wing MAVs fly at about 65 km/hr with chord Reynolds numbers between 4.5×10^4 to 1.8×10^5 at altitudes under 100 m.[26] The smallest

MAVs can fly in Reynolds numbers of only 10^3, and velocities of about 10 m/s.[27] Figure 2 shows the relationship between size and Reynolds number.[12]

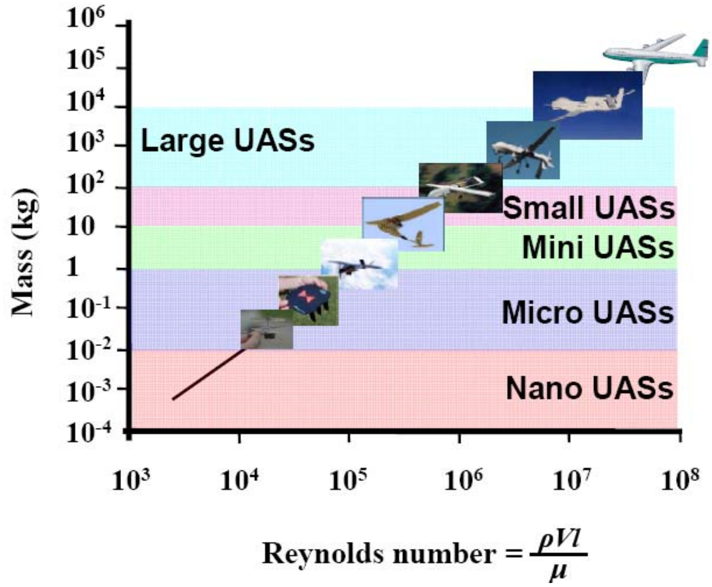

Figure 2. MAVs have small Reynolds numbers

Aerodynamic efficiency is measured by the lift-to-drag ratio. Most UASs have glide ratio greater than 10, but birds, insects, and MAVs are typically less than eight.[24,26] Contrary to well-understood large flight vehicles, lift-to-drag ratios actually decrease for smooth airfoils at Reynolds numbers less than 10^4.[56] Equation (6) shows a further inconsistency for low Reynolds number flight where the reference Reynolds number is $1.5\text{x}10^4$.[57] This states that within the Reynolds number regime of $1.5\text{x}10^4$, more lift is required when the ratio of profile drag to induced drag increases.[24] Profile drag is yet another unique parameter peculiar to low Reynolds number aircraft. Low Reynolds number airfoil data report slightly higher than expected profile drag coefficients of around 0.05.[58]

$$C_{L,\min power}{}^{2} = \frac{3C_{do}}{b}\sqrt{\frac{\mathrm{Re}_{ref}}{\mathrm{Re}}} \qquad \textbf{(6)}$$

In order to account for this disparity not predicted from scaling, new research is needed within this flight regime. At Reynolds numbers of about 10^5, laminar boundary layer flow tends to separate, and at 10^3, turbulence is important.[27, 54] Other problems unique to these MAVs include turbulent transition, separated shear layer and reattachment with separation bubbles, strong tip vortices, tip vortex destabilization, and roll instability.[26,27,55] Furthermore, the laminar separation bubble movement is nonlinear and difficult to predict.[24] Separation is typically only an issue near stall, but low Reynolds number MAVs exhibit separation during normal flight.[59] Due to the small size, positive static margins severely limit center of gravity travel. Because the flow is highly three-dimensional, computational analysis is intensive and does not yet lend itself to design at the conceptual level. Particle Image Velecometry (PIV) is an experimental videographic technique used to visualized fluid flow. Although PIV is often not appropriate for conceptual design, it is an invaluable experimental tool for investigating MAV flow phenomenon.[55]

Typical MAV aspect ratios are between one and two,[10] but the upper limit for MAVs can be expanded to five. Non-linear tip vortex effects are reduced and lift enhanced as aspect ratio increase, so MAV aspect ratios are only low due to limiting design criteria, such as the stipulation of a maximum dimension.[11] This is a design challenge since at aspect ratios less than two and Reynolds numbers less than 10^5, performance deteriorates. As aspect ratios drop below one, tip vortices produce most of

the lift. This process is highly nonlinear and has a destabilizing effect not predicted by existing thin airfoil data.[26]

II.3.2. Xwing

Xwing is an aerodynamic code that uses a two-dimensional boundary layer and a three-dimensional flow matching technique to produce better results through boundary layer iteration than by use of an inviscid solver alone. Efficiency is optimized through the ability to parameterize taper ratio, sweep angle, and twist. The key advantage to Xwing is that it is specifically designed for low aspect ratio wings and low Reynolds numbers. Although Xwing appears to be the perfect aerodynamic software tool for this application, it is currently undergoing validation and has not been released to the public for use.[53]

II.3.3. Athena Vortex Lattice (AVL)

Athena Vortex Lattice software has proven to be accurate and robust for diverse applications.[60, 61] AVL is a vortex lattice code developed by Drs. Mark Drela and Harold Youngren at the Massachusetts Institute of Technology. It focuses on the aerodynamic and flight-dynamic analysis of rigid body aircraft for a wide variety of configurations. It uses an extended vortex lattice model for lifting surfaces that can be combined with a slender-body model for the fuselage. It utilizes linearized equations of motion for any given flight conditions.[62] Outputs of AVL include various measures of lift and drag, stability and control derivatives, and Eigenmode analysis. AVL was used to accurately predict the stability of a design submitted to a DARPA MAV competition.[63] Wind tunnel tests of the GenMAV are currently in-work to validate the use of AVL with the GenMAV. No literature has been found that provides a lower limit on Reynolds number

for use with AVL, but the GenMAV wind tunnel tests will be able to validate AVL for similarly sized platforms. Preliminary results indicate that AVL accurately predicts flow conditions for low and moderate angles of attack.

II.3.4. Other Aerodynamic/Stability Codes

As of 1998, when computational analysis gained widespread use within universities at the graduate and undergraduate level, two separate sets of code were readily available for use. These included Digital Datcom and HASC (VORLAX). Digital Datcom automates the process of calculating stability derivatives. It is a built up analysis that runs in batch mode. Various cases with different speeds can be run at once, and induced drag is provided. Unfortunately, zero-lift drag is not very accurate, partly due to the inability to model inlets, canopies, and other small scale drag contributions. An improvement upon Digital Datcom was made with High Alpha Stability and Control (HASC) model, built on the well-established VORLAX code, which is a generalized vortex lattice code. This provides better vortex lattice analysis by taking into account discrete body vortex and wing vortex burst effects. While each has its specific strengths and weaknesses, only Digital Datcom provides static and dynamic derivatives. Both will provide force and moment coefficients and take into account ground effect.[64]

Gambit and Fluent Computational Fluid Dynamics (CFD) models were used to validate the Rochester Institute of Technology MAV that utilized an inverse Zimmerman planform and Selig low Reynolds number airfoil. This code adequately predicted overall drag and lift coefficients, but could not provide consistent induced drag coefficients.[11]

II.3.5. Airfoil Design

Specific design requirements often demand new airfoils. This is certainly the case for low Reynolds number MAVs. One group of researchers at the University of Illinois sought to create a design methodology for this purpose using PROFOIL, Eppler, XFOIL codes, and extensive experience.[65] The lower Reynolds number bound of this research appears to be 2.0×10^5, which is much higher than expected values for MAVs, but illustrates a process.

Rough design utilized PROFOIL, which is an inverse design method that uses the integral boundary layer approach. Flow characteristics can be set on the velocity distribution, and is constrained by the boundary layer and the thickness. In PROFOIL, the laminar boundary layer of the lower surface can be set to separate at the lift coefficient given on a desired lower corner of the drag polar. This requires the upper surface to adjust for the constraint. The analysis function of Eppler code was then used to determine inviscid velocity distributions by a third-order panel method, followed by an integral boundary layer method using inviscid velocity distribution for performance. Laminar separation bubbles were not included in the calculations. After Eppler, XFOIL confirmed the design with a viscous/inviscid linear vorticity second-order panel method joined with an integral boundary layer method.

The most common planform is the Inverse Zimmerman type, which has shown to be most efficient for low Reynolds number flight.[11] The crafts are typically flying wings with aspect ratios near 1.5.[2] Adequate control capability can be achieved through the use of leading edge sweep for wing dihedral, fins to improve lift through reduced tip vortices,

symmetrically operated elevators for pitch control, and asymmetrically operated elevators for roll control.[2]

II.4. Specific Power and Specific Energy

Batteries provide an excellent source of power and various chemistries allow for a customized direct current supply for nearly any use. Unfortunately, stored energy can be quite limited, especially at high power rates. Fuel cells, on the other hand tend to have an abundance of energy, but cannot reach the maximum power values of batteries. The following sections discus the power and energy attributes of batteries and fuel cells.

II.4.1. Batteries: High Specific Power

The primary performance properties used to compare electric power systems are specific energy and specific power. These are the energy capacity and instantaneous power normalized by weight. Unfortunately, selecting the best electric power source is not as simple as choosing the highest specific energy and highest specific power because these two performance metrics are proportional to discharge rate and not simultaneously maximized.

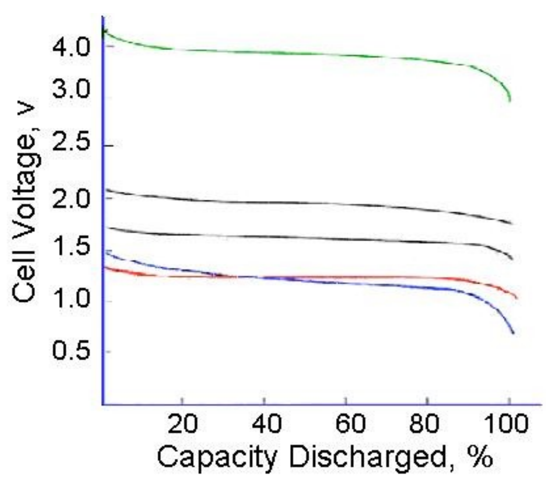

Figure 3. Cell voltage varies with type of battery and state of charge.

Batteries must be experimentally tested to truly understand the performance envelope. The electrical voltage draw correlates to the State Of Charge (SOC), as seen in Figure 3. The voltage also depends on the rate of discharge (C), and the capacity as shown in Figure 4.[70]

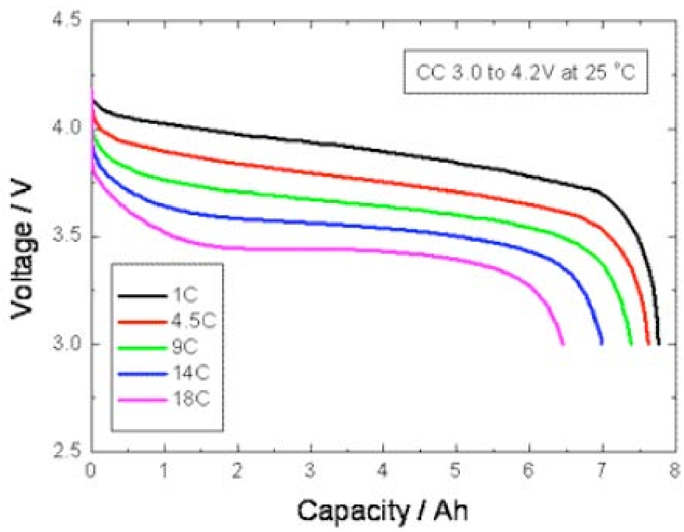

Figure 4. Cell voltage varies with discharge rate

The following terms are useful when evaluating battery terminology. Capacity is the amount of current load that the battery can provide over an hour before the voltage drops to an unusable level. This corresponds to a C rating of 1. Discharge rate ("C rate") is how fast the battery can be safely discharged. Another common use of the term is C/1 for a C rate of 1 and C/2 for a C rate of 2. Duration in hours is the inverse of the discharge rate.

In 2007, DARPA sponsored a Naval Research Laboratory (NRL) study that surveyed commercially available lithium polymer batteries that weigh less than 5 g. Since the study, availability and performance has greatly expanded, but some of their

results illustrate concerns regarding performance criteria. Table 2 lists the test battery specific power and specific energy results at low and highest discharge rates.[67]

Table 2. Survey of commercial small lithium polymer batteries (2007).

	Discharge Rate [C]	**Specific Energy** [W-hr/kg]	**Specific Power** [W/kg]	**Discharge Rate** [C]	**Specific Energy** [W-hr/kg]	**Specific Power** [W/kg]
1	1	127	123	12	32	1147
2	1	145	137	20	119	2383
3	1	159	157	12	130	1637
4	1	108	97	18	60	1394
5	1	96	25	15	68	1511
6	1	109	95	20	85	1700
7	1	138	119	20	113	2125
8	1	109	115	12	25	1040
	Average	124	109	16	79	1617

The biggest take-away from this table is a concern well known to those familiar with battery technology: performance metrics vary greatly depending on discharge rate. When the discharge rate is 1, the specific energy value can reach the maximum because the battery can expend nearly all the stored energy. When the discharge rate is 20, the actual energy available is much less, but the specific power can be maximized. In fact, these data display a 9 to 20-fold increase in power at fast discharge rates compared to power available during a slow discharge coupled with a 60% decrease in specific energy. Ragone plots illustrate this relationship. The user must have an appreciation for this tradeoff and choose specific energy and specific power values for the same discharge rate.

Batteries do not downscale linearly. Nearly the same amount of packaging material such as wrapping and insulation is needed for a smaller battery while the amount

of energy producing material decreases. This ratio introduces a weight penalty as batteries are downscaled. Utilizing the overall battery performance specifications of the three "best" batteries from the NRO study, realistic values are 120 W-hr/kg for specific energy and 2050 W/kg for specific power. These average values are for the highest discharge rates. At high discharge rates, the specific energy is lower and the specific power is greater than what one would expect from a Li-Po battery. The tradeoff is duration. These small batteries can provide power and energy at these rates for only a few minutes. These data and other generalized battery specific energy and specific power performance specifications are summarized in Table 3.

Table 3. Various battery characteristics found in literature.

Type of Battery	Specific Energy [W-hr/kg]	Specific Power [W/kg]	Ref.
Battery (in general)	200	-	15
Li-Ion Polymer (129 W-hr)	160	1203	68
Li-Ion Polymer	200	2800	42
Li-Ion Polymer	148	-	14
Li-Ion Polymer	150	4000	withheld
Li-O_2	1000	50	withheld
Li-Po (small, high C avg)	120	2050	67
Li-Po (current technology)	130-200	2800	69
Li-Po (future technology)	290-420	2500-5900	69
Li-S	350-400	-	42, 24
Nickel Metal Hydride	80	1000	42
Nickel Cadmium	60	150	42
Sodium Borohydride	7100	-	42

It has been shown in the previous sections that maximum specific energy and maximum specific power are not independent. Furthermore, the rate of discharge affects available power and energy as well. The Ragone plot captures this complex relationship.

A Ragone plot specifically created for these types of lithium polymer batteries used in MAVs is shown in Figure 5.[72]

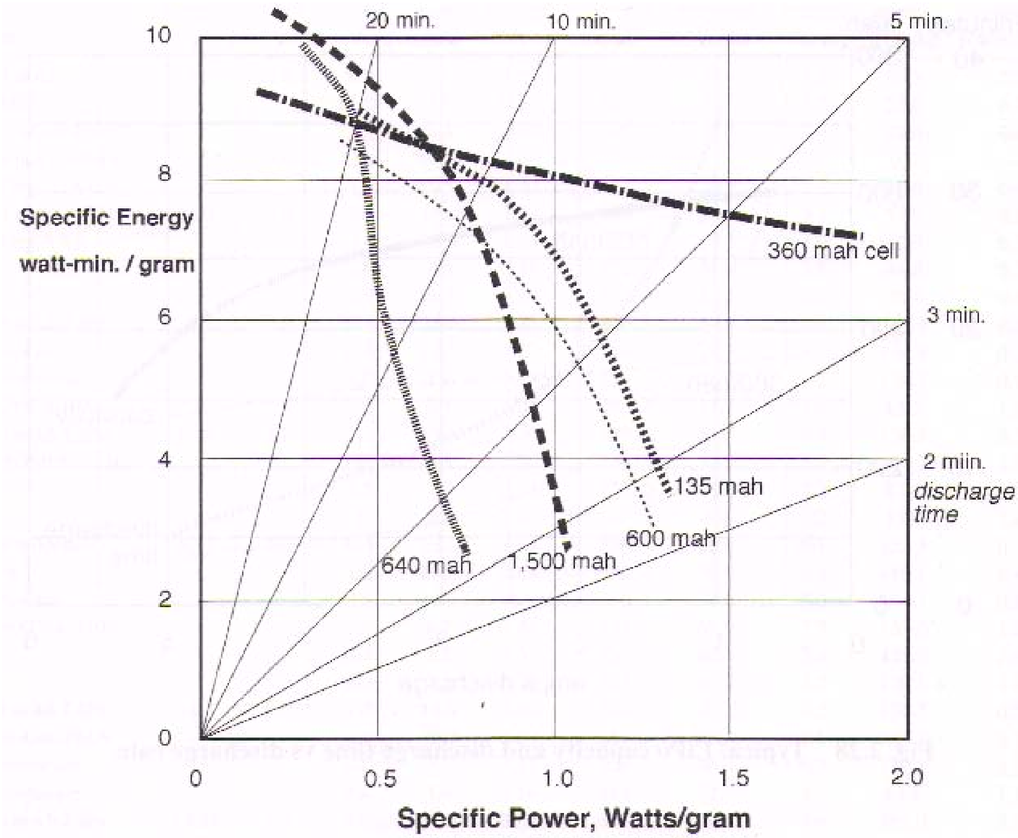

Figure 5. Ragone plot of Li-Ion batteries used in MAVs.

New battery chemistries continually improve possibilities. Lithium sulfur batteries are expected to have specific energy in excess of 400 W-h/kg.[24] The newest and most promising batteries combine a high specific energy source and a high specific power source to produce a carbon monofluoride silver vanadium oxide (Li/CFx-SVO).[79]

II.4.2. Fuel Cells: High Specific Energy

Fuel cells are not new. They have been around since the mid 1800's.[73] The US space program used them on the Apollo and currently uses them for water on the Space Shuttle.[74] Between high energy prices and more concern for the environment, fuel cells

have once again become attractive. High specific energy as compared to batteries is the main benefit. For example, fuel cell systems using compressed hydrogen have a 28% higher energy to weight ratio than advanced research lithium sulfur batteries.[14]

Simply put, fuel cells convert the chemical energy of fuel and an oxidant into electrical energy that can do work. Electrons break away from a chemical species. The remainder is in a charged state that passes through a membrane before recombining with the oxidant at the cathode.[13] The membrane catalyst, often made of precious metals, motivates this process. Plates of Membrane Electrode Assemblies (MEAs) fit together in stacks not conceptually different then battery cells.

II.4.2.1. Types of Fuel Cells

One of the most simple classifications of fuel cells are those that need pure fuel, and those that need to process an impure fuel to make it useable. The preprocessing technique is called reforming. Reforming separates the hydrogen into a usable form not contaminated by other byproducts. Generally speaking, due to the small size of MAVs, the current state of technology does not allow for reformation in flight. This leaves two primary fuel cell candidates for MAVs: Polymer Electrolyte Membrane Fuel Cells (PEMFC), and Direct Methanol Fuel Cell (DMFC).

Polymer Electrolyte Membrane Fuel Cells (PEMFCs) take advantage of a proton conducting membrane between two platinum impregnated porous electrodes. They operate at a low temperature of around 80° C and a pressure of between $7.0\text{x}10^4$-$7.0\text{x}10^5$ Pa gauge. Unfortunately, at these low temperatures, carbon monoxide poisons the platinum catalysts that sustain the reaction. Even at only 10 parts per million, carbon monoxide negatively affects performance. Another challenge is maintaining the right

amount of humidity. Resistance is reduced when the membrane is wet. The flow of water through the cell is an osmotic process and dependent on the current and material characteristics of the membrane and electrodes. As current density increase, the osmotic process decreases, reducing cell performance. These fuel cells are very much temperature and pressure dependent.[35]

DMFCs are an attractive option to pure hydrogen PEMFCs because the fuel does not need to be reformed prior to use. Liquid transport is easier than gaseous, but methanol crossover and high overpotentials are challenges.[35]

II.4.2.2. Fuel Cell Losses

Real fuel cells are subject to irreversibilities that reduce the actual voltage from an ideal voltage dependent upon the amount of current drawn. The main losses are activation polarization (reaction rate loss), ohmic polarization (resistance loss), and concentration polarization (gas transport loss). Losses can be minimized, but usually require operation at a higher temperature, which causes other design tradeoffs.[35]

Polarization is a term that is used for irreversible losses. These losses can be summed within the static polarization curve. A polarization curve that includes the BOP can be used to predict performance.[4] In general, the fuel cell performs best when it delivers steady low-load power. Efficiency decreases with increases loading due to mass transfer and ohmic losses.[4, 75]

The complexity and multidisciplinary nature of the fuel cell system explains why only limited design models are available.[50] Many fuel cell disciplines, to include electrochemistry, heat transfer, mechatronics, and controls need further research with special attention to airframe system level integration.

II.4.2.3. Fuel Cell Specific Energy and Specific Power Performance

As complicated specific energy and specific power metrics are for batteries, they can be even less clear for fuel cells. Power output follows a similar polarization pattern of decreased voltage and decreased energy capacity at higher current draw, but the term "discharge" leads to confusion. Whereas a battery has a certain amount of stored electrochemical energy that discharges over time, a fuel cell is an electrochemical conversion device that could theoretically continue to provide energy so long as the fuel is available.

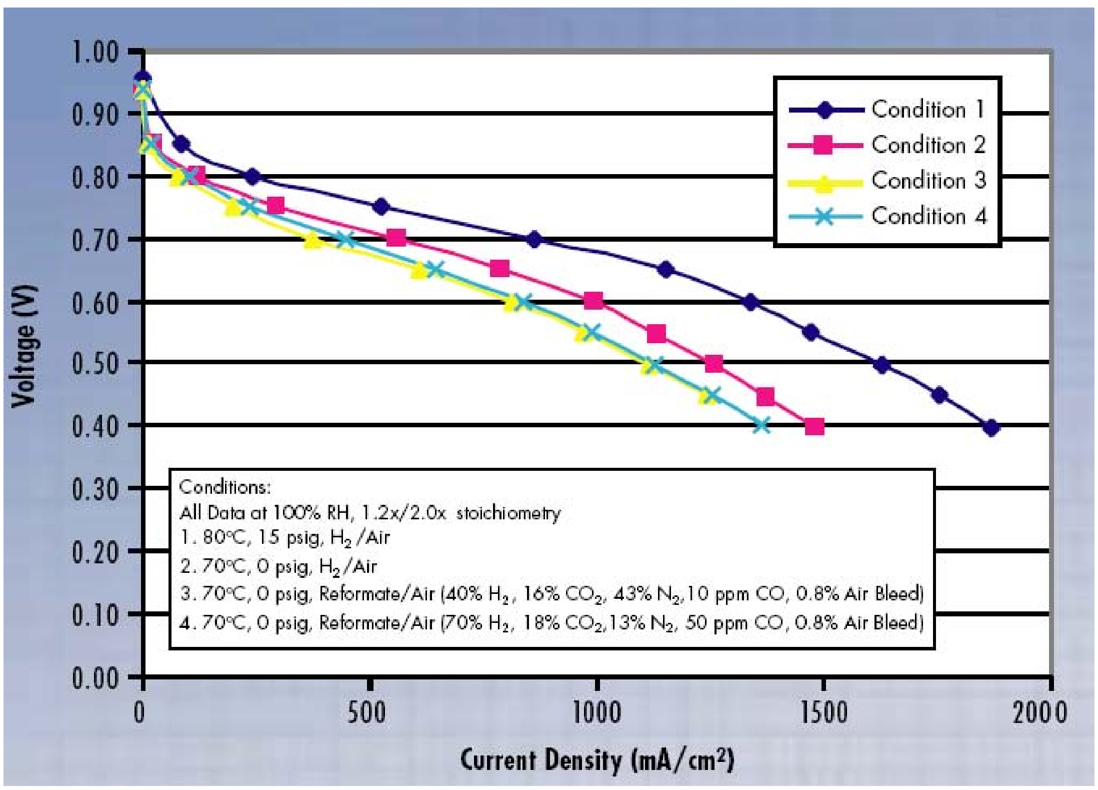

Figure 6. Fuel cell stack voltage is dependent on current draw.

Power available from the fuel cell is a function of the number of individual cells and the power draw as shown in Figure 6.[70] Best operating efficiency is achieved when

current draw is low, and within the middle linear region. Fuel cells perform poorly with fluctuating current demand. Conventional wisdom dictates that the fuel cell shall be operated at the average power demand level, and this effort will test that theory. Results are provided in section IV. Specific power can then be determined by normalizing the output power by the weight, which includes the BOP.

Unlike the rather straightforward means to determine specific power of a fuel cell, the specific energy is technically infinite, and therefore, can be an unreliable specification unless one is clear about how the specific energy was determined. The amount of reactant fuel available is the only constraint on the amount of energy that the fuel cell can provide. This is analogous to the internal combustion engine used in automobiles where the weight of the gasoline must be included in a system level specific energy calculation. Fuel cells generally operate on cartridges. Therefore, a meaningful value of specific energy for a fuel cell must be system level, include the BOP, and be clear about how much fuel is included (how many cartridges). It can be very difficult to ascertain how much of the BOP is included within an energy or power per unit mass specification because no common convention exists. Table 4 gives a brief summary of available values found in the literature. The numbers vary wildly between various subtypes of fuel cells, storage solutions, and BOP makeup. The extremely small value of specific power given by Ref. 42 is the result of a system built to provide a low level of power for a long period, and does not indicate a maximum capability. A sample Ragone plot that displays multiple electrochemical sources is shown in Figure 7.[71]

Table 4. Sample of fuel cell energy/power found in literature.

Type of Fuel Cell	Qualifier	[W-hr/kg]	[W/kg]	Ref.
FC, Ragone plot (in general)	up to	1000	100	66
FC (in general)	greater than	800		15
PEMFC, H2 gas		1000		19
PEMFC, Liquid H	greater than	10000		19
Given system in paper (PEMFC)		7.1	52	19
PEMFC (PUMA)		350	500	3
FC, H2 gas, automotive			245	20
FC, Liquid H			442	20
Given system in paper (PEMFC)		450	0.07	42
FC, automotive, scaled		1240		14

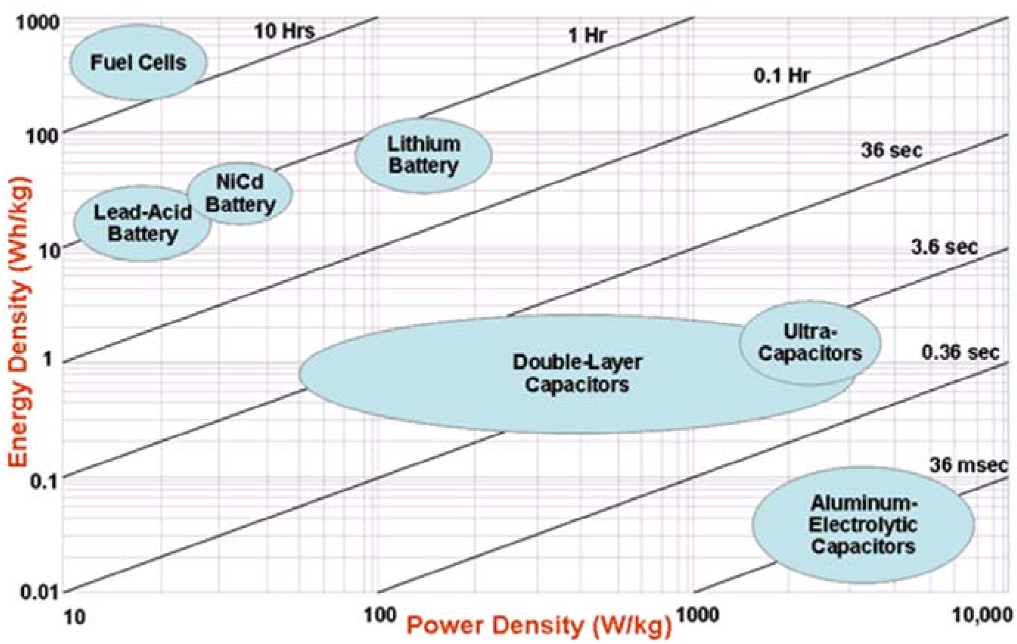

Figure 7. Ragone plot for various electrochemical storage devices.

II.4.2.4. Other MAV Fuel Cell Alternatives

More fuel cell varieties are commercially available then ever before. This research focuses on MAVs, so fuel cell types that do not require reformation such as PEMFCs and DMFCs are the most likely candidates. Protonex, a company that makes PEMFCs, can achieve up to 500 W-hr/Kg at relatively low temperatures of 20-90 °C.[3,13] A battery-PEMFC hybrid with battery charging capability can have a system level specific energy of over 800 W-hr/kg, which beats a battery alone, even at low charging efficiencies.[15] In addition to PEMFCs, possible MAV alternatives are DMFCs and Reformed Methanol Fuel Cells (RMFC). Samsung Electronics recently announced a 25 W DMFC capable of 550 W-hr/kg that is rugged enough for portable use by soldiers.[76] Another portable soldier power effort utilizing RMFCs was announced by UltraCell that can supply 25 W and is expected to provide 800 W-hr/kg.[3] It has already passed military safety tests for variations in altitude, temperature, humidity, dust, and vibration. Lastly, Solid Oxide Fuel Cells (SOFC) are mentioned because they are likely candidates for larger systems, such as UASs. SOFCs operate at 600-1000 °C, so they are not appropriate for small applications such as MAVs, but have found use in ground power plants. SOFCs may be used for large aircraft in the future because waste thermal energy can run other processes, less pre-reformation is necessary, they are more tolerant to impurities such as sulfur, and because they can use carbon dioxide as a fuel.[13]

II.4.2.5. Balance of Plant

The Balance of Plant (BOP) consists of any subcomponents to the fuel cell system required for the fuel cell to generate power. The BOP could be called the fuel cell "overhead." BOP designs and sizes require MDO in much the same way as an aircraft

would. Compressors are often needed to provide adequate air supply. Humidity and temperature must be controlled. Tradeoffs exist for each subcomponent. Activation, mass transport, and overpotential losses are high when fuel cell temperature is too low. When the fuel cell temperature is too high, greater than about 65 °C, the fuel cell self-humidification is compromised, and the fuel cell membrane cannot conduct at full performance.[76]

One primary component of the BOP is the reformer. This is required any time pure hydrogen is not supplied to the PEM fuel cell stack. Furthermore, the reformation process typically releases carbon dioxide, which may negate some environmental benefits.[13]

II.4.2.6. Storage

Any fuel cell system must take into account the volume necessary to store the required energy. This research does not attempt to integrate volume requirements into the MAV conceptual design, but these requirements are key limitations. The simplest dichotomy is storing fuel as a compressed gas verses a cryogenic liquid. Gas storage is simplest to use, but requires the most volume.

Hydrogen is often a better choice of fuel because of its ready-to-use purity. Liquid hydrogen, however, carries far less energy than other liquid fuels, such as kerosene, but costs an additional 40% in volume. Energy verses volume tradeoffs are illustrated in Figure 8. Heavier fuels, such as methanol or liquid methane weigh more and need to be reformed.[13]

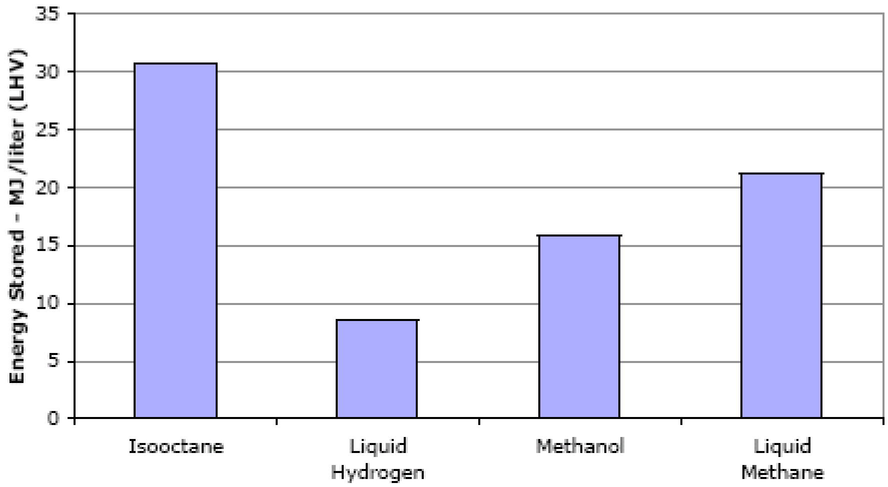

Figure 8. Lower heating value energy density of select fuels.

II.5. Energy Management of Hybrid Systems

Minimum fuel consumption control strategies are effective for internal combustion engines, but less understanding exists for pure electric-powered aircraft.[80] Recent research indicates that power minimization planning for hybrid-electric aircraft is somewhat counterintuitive. In fact, neither hybridization, nor flight path optimization extends endurance when the batteries must maintain a certain state of charge. As compared to the climbing and gliding flight path required for internal combustion engine aircraft, electric aircraft are optimum at straight and level flight. Because fuel cells operate most efficiently at a constant load, the overall efficiency can improve if the fuel cell is decoupled from the total power requirement. It is suggested to separate the climb rate constraints from endurance constraints. An example of this strategy is to use batteries for take-off and the fuel cell for cruising, which would produce longer endurance. With no need to recharge, the batteries would become unused payload after take-off.[4] Because fuel cells can reach specific energy levels of up to five times that of

batteries, long endurance flights might be optimal without carrying any batteries.[4] The optimal degree of hybridization between batteries and fuel cells would differ for different missions, but an initial investigation of a large, 700 W, fuel cell-battery-powered aircraft system designed for endurance showed that a fuel cell sized to provide between 60-80% of the system power could optimize endurance.[4]

The hybridization will likely be necessary for any MAV utilizing a fuel cell due to relatively high power fluctuations caused by stability and control requirements. The conceptual design tool described here could answer these types of power management questions.

II.6. Propeller and Motor Models

Both the propeller and motor are major subcomponents to any electrically propelled system. In system-level design, two separate, but dependent models are required. Fortunately, a software program called QPROP assimilates two highly developed models for the propeller and the motor.

II.6.1. QPROP

QPROP is a performance model for propeller-motor combinations. It uses a relatively sophisticated and accurate propeller aerodynamic model and a generalized motor model.[81,82] The propeller aerodynamic model employs an advanced blade-element and vortex method. The QPROP documentation says that the additions to the method are based on the work of Larrabee and correct for self-induction to improve accuracy at high disk loading. The blade airfoil lift characteristic is assumed a simple linear $C_{L\alpha}$ line with C_{Lmax} and C_{Lmin} stall limiting. The profile drag characteristic is a quadratic function of

C_D dependent on C_L, with an approximate stall drag increase and includes power-law scaling with Reynolds number. Inputs to QPROP are shown in Figure 9.

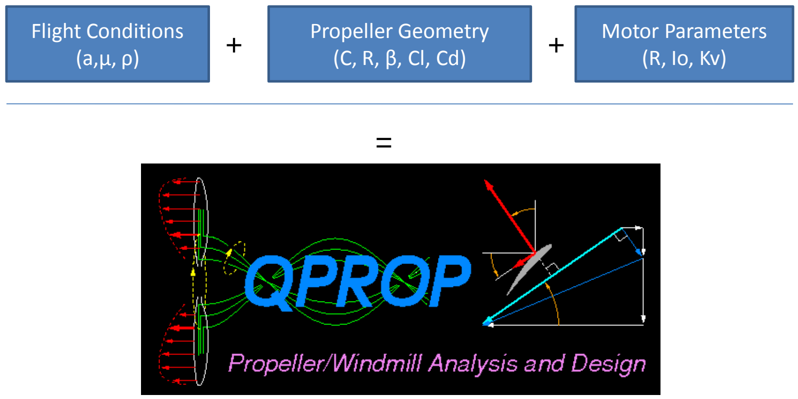

Figure 9. QPROP inputs.

QPROP internally optimizes four parameters: RPM, pitch rate change, thrust, and velocity. Outputs of QPROP are speed, RPM, pitch rate change, thrust, torque, shaft power, voltage, current, motor efficiency, propeller efficiency, advance ratio, thrust coefficient, torque coefficient, slipstream velocity increment, electrical power, propeller power, power weighted average of local C_L and power weighted average of local C_D. QPROP must have at least two inputs of the four key parameters, or it will not run.

II.6.2. Motor

Brushless motors are now available that provide significantly more power and better efficiencies than brushed motors. Despite the requirement of an electronic speed control device, the weight difference is negligible.[2] Researchers at a Swedish university recognized that the propulsion system, to include the battery, motor, controller and propeller is key to optimizing MAV capabilities. Focusing primarily on the battery, they discovered that standard equations for brushless DC motors could be improved upon,

manufacture's data should not be trusted, and the motor controller model is complex. However, this study would be very useful to someone seeking to improve the motor model characteristics based on more vigorous and faithful motor tests.[85] Any motor model can be incorporated into QPROP. Since QPROP analyzes the propulsion system performance, it must have both a propeller and a motor as input files and cannot run one without the other.

II.7. Software Linkage and Model Center

Historically, aircraft design relied upon groups of specialists who iterated on a design until all or most of each group's requirements had been satisfied. In the computer age, with software, a single person can orchestrate the effect of multiple groups' efforts. Because engineering is specialized, multidisciplinary design usually requires the interaction of several pieces of software. When each software piece was not originally designed to interact with others, it can take extensive computer programming skill to enable these interactions. A program called Model Center provides a means for inter-software communication and offers a graphical user interface for model manipulation.

Other researchers are focusing not only on the conceptual design of aircraft, but also on the collaboration among geographically separated people.[86] A conceptual design environment called Distributed Analysis Modeling Environment (DAME) has previously been developed to quickly incorporate multiple models not necessarily collocated so that numerous contributors could collaborate on a multidisciplinary project. Due to networking challenges, efforts such as these would prove tangent to this paper's goal, but is notable because architecture concept verification employed Model Center with many

subcomponents for the purpose of air vehicle design. DAME used Flight Optimization System (FLOPS) as the airframe model, which is limited to a single set of input conditions to generate flight performance metrics. In this case, multiple mission segments such as climb, cruise, loiter, and descent could not be independently inputted into FLOPS, so various mission segment input deck configurations were written into Model Center so that the system could generate a more dynamic flight performance profile. For the project mentioned here, a paired layering approach was used to keep subcomponents modular. The concept required interfaces in the form of script files to relay information between various components. Despite a slight increase in complexity, the user can effectively "swap out" components with similarly configured components with relative ease.[87] In addition to FLOPS and Model Center, some other optimization codes used for this purpose are NEWSUMT-A, OptdesX, Mathworks Matlab, and Microsoft Excel Solver.

Model Center allows a variety of file types to be linked together to form input/output relationships. It supports popular utilities such as Mathworks Matlab and Microsoft Excel with plug-ins, and batch mode executables such as QPROP and AVL with user-created file wrappers. These require some knowledge of MS-DOS, server based architecture, Visual Basic, and VBscript. Any number of component files can be linked together to build a single program model. Then, built-in capability such as trade studies, Design Of Experiments (DOE), or any other user-determined analysis can be run on the single interconnected Model Center model. Note that, like any model, the Model Center model must contain component models that are dynamically and geometrically similar for realistic results. Otherwise, individual components within the model must be

revised to support a separate analysis. A more detailed description of Model Center's capabilities can be found in Chapter 3.

Chapter 2 started with an introduction to MAV and UAS platforms that either demonstrated alternative power systems such as fuel cells and hybrid-power sources or illustrated the MDO concept. When many disciplines must collaborate to produce new technology that might not operate the same as a scaled up version, MDO becomes critical. Various aerodynamic and propulsion codes were reviewed along with important aspects of batteries and fuel cells. Lastly, the Model Center framework was introduced as a means to tie all the pieces together.

In Chapter 3, the operational details of each chosen component will be explained. AVL, QPROP, and airframe power required models will be thoroughly examined. The importance of subcomponent models for fuel cell and battery performance will be shown as it pertains to the conceptual design tool. Lastly, an introduction to Model Center's built-in trade study and optimization techniques will be discussed.

III. Methodology

Within the methodology chapter, all components and interactions within the model will be explained. Relevant theory and assumptions will be discussed and modes of operation will be described.

III.1. Conceptual Design Tool Overview

Various models and components are integrated into a single interface to assist the power systems engineer in the implementation of an optimum power system. In addition to the power system focus, aerodynamic performance available from AVL blended with the propeller and motor analysis of QPROP offer the ability to separate each component for a tailored sub-optimization if desired.

The conceptual design tool developed here is meant to be used in one of two modes of operation. The first mode is set within the context of a single mission segment. The user supplies the flight conditions by specifying altitude and velocity, and the program calculates the required power that must be supplied from the power system. The organization diagram of this mode of operation is shown in Figure 10, which also is the Model Center component interface. It is the inner loop used for any mission segment.

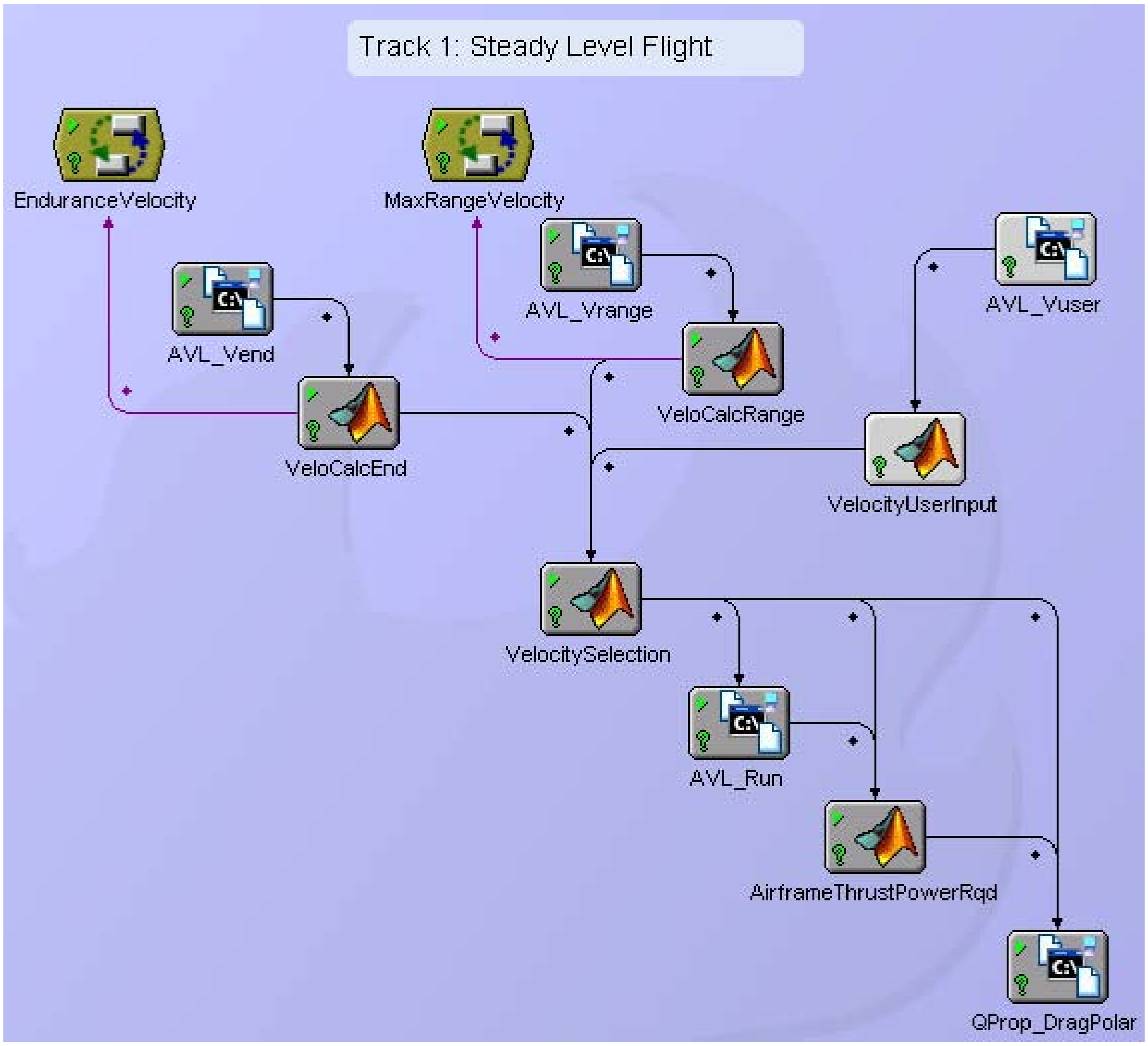

Figure 10. Single iteration inner loop for power-required calculation

The second mode of operation creates a sortie of five mission segments linked in series. The user describes what altitude, and climb rate each segment will have, along with the type of mission segment such as cruise and loiter. Velocity can be either a specific speed or the speed that requires minimum power or minimum thrust for endurance or max range, respectively. A simple baseline mission profile is illustrated in Figure 11.

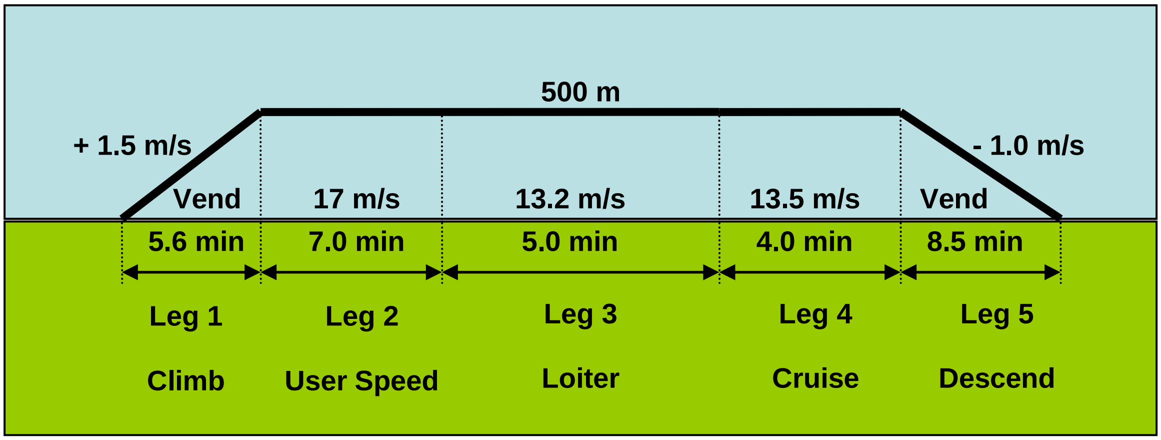

Figure 11. Baseline mission profile.

The program runs each prescribed mission segment and calculates average power required for each segment. An additional power measurement is taken at the midpoint of the climbing and descending mission segment. Once all the average power values are logged, the user-inputted specific energy and specific power are checked against the mission power required. An aggregate power system mass is calculated. A graphic of the outer loop used to simulate this series of mission segment points is depicted in Figure 12. The component called Threshold Values is the gradiaent optimizer that can be used to analyzie the optimal split of power components.

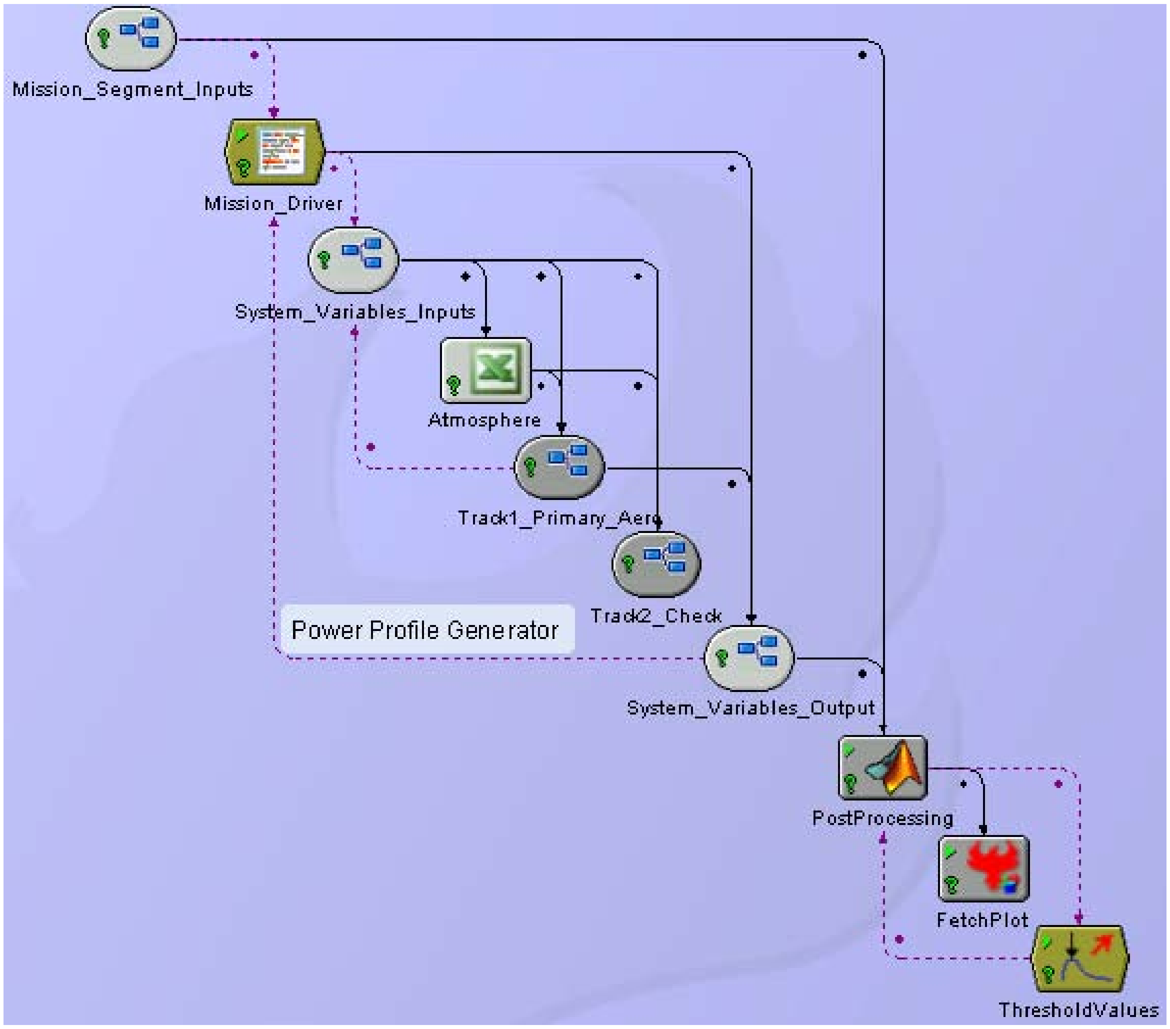

Figure 12. Outer loop used to run mission segment simulations.

This process can be iterated and optimized on any design variable or variables to answer questions pertaining to feasibility or to determine requirements for a variety of changes. Each component of the model will be explained in detail in the following sections.

III.1.1. Model Center

Model Center is the integrator used to connect all software internally and provides a graphic user interface from which variable values and attributes can be modified. It

also orchestrates the execution sequence of each component. Trade studies, DOEs, and optimization can be performed within the Model Center environment.

Software interfaces with Model Center through various means depending on the type of software. Popular applications such as Matlab, Mathcad, CAD programs, and Excel have a specially designed interface called a plug-in. The plug-ins are written by Phoenix Technologies, the company that owns Model Center. These plug-ins have version numbers like any other program so that they can keep up to date with both Model Center changes and third party software changes. The plug-in provides a Graphic User Interface (GUI) for the user to simplify software connection, and in most cases, makes connections easier. The model used in this study takes advantage of Matlab and Excel plug-ins. The main way Model Center connects to other programs is by way of the Analysis Server. Analysis Server is a complementary program that Model Center uses to manage connections with batch mode executable components. It runs in the background and is separate from Model Center. Batch mode applications that run with an executable program from the command window require Analysis Server. On some machines, it runs as a service in the background, but must be started independently on others. In addition to batch mode components, scripts also use Analysis Server. Scripts can be written in a variety of languages to include VB, VBScript, C++, and JAVA, directly within the Model Center environment or outside. Examples of the components used for this effort, along with explanation are included in Appendix C. All Model Center components are typically saved and written to within the same file directory as Analysis Server. The user must have read/write access to the drive where Analysis Server is installed.

Once all of the components are connected within Model Center and the variables are defined, the entire model can be run with a single button. Model Center usually knows when to run each component automatically by starting at the bottom (output) of the model tree. It checks for validity of each end variable. Validity is whether the variable is up to date. If the end variable is invalid, or stale, Model Center knows to reach back to the previous component. It does this all the way until it reaches the user input variables at the first component. This backward approach is ideal for most models. When a model does not operate in a linear fashion, as a complicated model might, the Script Scheduler can determine how to run the model. The Script Scheduler is a script written to explain to Model Center exactly when to run each component. The Script Scheduler was initially used for the multi-segment mission runs, until a more powerful technique that used a driver was discovered.

In addition to the Script Scheduler, other components can orchestrate a model. Sometimes certain tasks must be performed until a condition is met. For these applications, a driver is needed. A driver was used here to run the mission profile. In order to run the mission profile, reference variables within the driver tracked and changed input conditions for each iteration so that they either matched the previous iteration or were as described by the driver script. After each mission segment ran, data was collected and saved for post processing and plotting.

III.2. Detailed Description of All Subcomponents

This conceptual design tool integrates various components within the Model Center environment. The major models represented by distinct software components are

AVL and QPROP. Other models based on simple sets of equations have been written into Matlab script for execution with the MatLab Plug-in. These include models for airframe power-required, fuel cell, battery, power system management, and mass determination.

III.2.1. Aerodynamics Model: Athena Vortex Lattice (AVL)

As previously introduced, AVL is a vortex lattice method of aerodynamic analysis and was chosen for this endeavor because it fits well at the conceptual design level and has been proven acceptable for the large MAV size. Some inherent limitations come with the use of any vortex lattice method and will be briefly summarized in this section. Despite the limitations, AVL is a very good choice among various other possibilities. The fundamental equations used in its analysis and how this analysis is integrated with the MAV conceptual design model will be explained.

Vortex lattice methods are a subset of Computational Fluid Dynamics (CFD), and can be summarized as the modeling of a series of infinitely long, discrete vortices along the lifting surface. By integrating forces of each vortex, lift and induced drag are calculated. Geometry and boundary conditions strongly influence the calculation's validity. The author of AVL reminds users of some basic vortex lattice rules generally required for good data, but also explains how the program accounts for the less than expert user in the help file "avl_doc.txt," available from the software website.[62] First, trailing vortex legs must not pass close to a downstream control point. If they do, they should have the same spacing. Second, spanwise vortex spacing should not have discontinuous changes in spanwise strip width. More strips are required at points of geometric change such as dihedral and chord breaks, control surface ends, and wing tips.

Chordwise strips of narrow width are required on control surfaces along hinge lines.[62] These guidelines are helpful when creating a vortex lattice structure used in conceptual design, but it is recommended to corroborate a new vortex lattice structure design with someone familiar with good CFD methodology. Figure 13 shows the GenMAV AVL model.

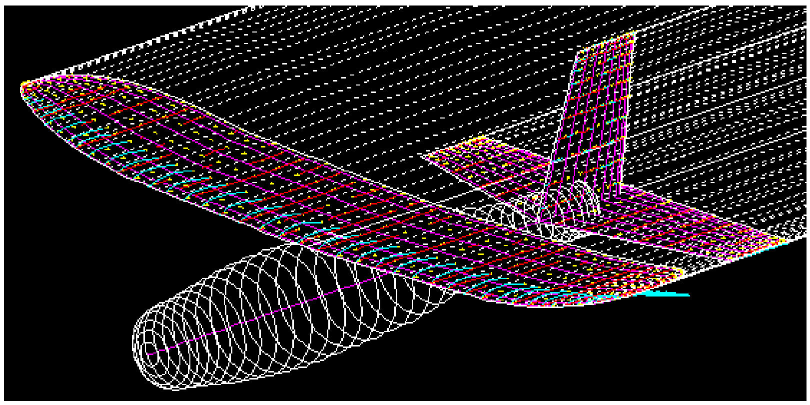

Figure 13. GenMAV AVL model illustrating mesh, camber, and trailing vortices.

III.2.1.1. AVL Theory of Operations

AVL first calculates the lift/span loading, *L'*. By means of the calculated circulation, Γ, the local lift coefficient, c_L, is determined by Eq. (7).[62]

$$c_L C_{ref} = 2\Gamma = 2\frac{L'}{\rho V^2 C_{ref}} \quad \textbf{(7)}$$

Note that c_L is only a measure of chordwise loading intensity in the streamwise direction. Boundary layer development depends on more than just the streamwise pressure gradients, so c_L may not be a good indicator of local stall. Therefore, AVL should only be used for low angles of attack where stall is not suspected.

In addition to lift, the total drag coefficient is found by cumulative surface force integration over each lifting aerodynamic surface, and induced drag is calculated from the

wake trace in the Y-Z plain far downstream. Parasitic drag, $C_{D,0}$, cannot be calculated by AVL and is a user supplied input. The span efficiency factor, *e*, is given by Eq. (8).

$$e = \frac{C_L^{\ 2} + C_Y^{\ 2}}{\pi C_{D,i} AR} \tag{8}$$

The value of span efficiency depends upon loading. Induced drag, $C_{D,i}$, is the AVL internally-computed Trefftz Plane drag coefficient calculated from the wake trace in theY-Z downstream plane. C_D, C_Y, and C_L are each the sums of forces in the respective X, Y, and Z directions each normalized by the product of dynamic pressure and planform area. *AR* is the aspect ratio found with reference values.

AVL has a simple interface for making flight condition changes during steady flight. For any single user input, AVL tries to balance Eqs. (9) – (13),

$$V = \sqrt{\frac{2mg}{\rho S C_L \cos\phi}} \tag{9}$$

$$R = \frac{V^2}{g\tan(\phi)} \tag{10}$$

$$\varpi = V/R \tag{11}$$

$$q = \varpi \sin(\phi) \tag{12}$$

$$r = \varpi \cos(\phi) \tag{13}$$

where an arbitrary bank angle is ϕ, and is positive right wing down. Mass is *m*, gravity is *g*, air density is ρ, turn radius is *R*, and turn rate is ω. Pitch rate, *q*, is positive nose up, and yaw rate, *r*, is positive for a right turn.

Mass, gravity, and dynamic pressure are predetermined from either the ".mass" or ".run" input files, or can be directly entered. When velocity is changed, the coefficient of lift must also change for equilibrium. Note that AVL will fail to converge at

unreasonably high C_L values required for very slow velocities, or high angles of attack. This is not a direct measure of stall, but the user is required to recognize the onset of stall and diligently avoid this regime.

AVL is capable of more than described here, such as Eigenmode Analysis and looping flight, but these functions were not ultimately employed within the conceptual design model.

III.2.1.2. AVL Operation: Inputs

AVL requires several inputs. First, the specific geometry of all lifting surfaces must be described. All spatial locations, chord lengths, control surfaces, and airfoils are written to a main file that essentially describes the geometry. Basic information such as air density, Mach number, velocity, center of gravity, wing reference area, chord reference length, and span reference length are inputted here. Default profile drag coefficient ($C_{D,p}$ or $C_{D,0}$) added to the geometry can also be included, but is assumed zero if absent. AVL handles compressibility with the Prandtl-Glauert compressibility factor shown in Eq. (15), where M is freestream Mach number.

$$\frac{1}{\sqrt{1-M^2}} \tag{14}$$

For slow flight, Mach less than 0.2, the AVL manual suggests that the Mach number could be set to zero to save computational time. The Prandtl-Glauert corrections allow AVL to provide reasonable results up to Mach 0.6. Furthermore, the users guide suggests that if the fuselage is not expected to contribute much to the lifting distribution, it could be omitted in the AVL model.

Exact dimensions of the airfoils are included within the geometry file. This data can either be transcribed from an airfoil reference, or can be calculated by a secondary, well-respected MIT software program called XFOIL. The main purpose of XFOIL is to calculate real aerodynamic airfoil coefficients for unique airfoils.[90] XFOIL was used to analyze the GenMAV lifting surfaces. The airfoil geometry file consists of non-dimensionalized X and Y coordinates from the trailing edge to the leading edge, and back to the trailing edge.

In addition to airfoil data within the main geometry file, information about the desired lattice structure must be provided. This includes the number of chordwise horseshoe vortices placed on the surface and the chordwise vortex spacing parameter. These spacing parameters describe how the vortex lattice panels are discretized within the geometry, and determine the spanwise and lengthwise horseshoe vortex node distributions. The underlying AVL vortex lattice solver is only concerned with the overall collection of the individual horseshoe vortices, so each surface need not be described in great detail. Only the minimum detail for clean interpolation between sections is required. The geometry file, genmav2pt2.avl, can be seen in Appendix C.1.e. In addition to the main geometry file, the GenMAV model includes 11 different airfoil files that describe the changing shape of the wing from root to tip. These are documented in Appendix C.1.g. Although the ability to see the lifting surfaces was not enabled through Model Center, when the GenMAV is run within AVL independently, a pictorial view of the MAV can be generated by AVL. Whenever a new model is written for AVL, it is highly recommended to view this graphic to ensure the encoding matches the desired MAV shape. AVL's graphic of the GenMAV was shown in Figure 13.

Once the geometry is established, AVL requires a mass file. The mass file provides AVL all mass data, moment of inertia data, cross moments of inertia data and Center of Gravity (CG) data for each subcomponent. The aircraft may be broken up into as many subcomponents as desired. The mass file also tells AVL what units to use during trim analysis. If units are not metric, scale factors must be included within this file. The mass file, genmav2pt2.mass, can be found in Appendix C.1.j.

The last important file is the run-case file. Although all options for the run case can be input via the AVL command prompt, it is much more convenient to write a file to track and vary each case, or to simultaneously load multiple cases. If present, information within the run case file overwrites any conflicting mass, moment of inertia, drag, and CG data from other files. This provides a single location to change multiple parameters without interfacing with menus. All parameters may be put into the run-case, but since certain parameters may be held as constraints, such as for trimmed flight, some variables will be changed internally. The parameters that may be input as constraints are alpha, beta, roll rate, pitch rate, yaw rate, roll moment, pitch moment and yaw moment. During program execution for trim, the user must review each parameter to check for feasibility. For example, a common example of an impossible case might be where C_L wasn't constrained, so AVL called for a C_L that exceeded the airfoil's capability, which will happen when velocity is slower than stall speed. Reasonable inputs were confirmed by inputting desired conditions, running the case, and then using the converged variable values as a basis for single parameter changes. Common parameters for the run case are the above constraints, as well as $C_{D,o}$, bank angle, elevation, heading, Mach, velocity, gravity, turn radius, a load factor, and mass. Despite guidance in the user guide to the

contrary, $C_{D,o}$ was only changeable through the run file, and not through the AVL geometry file. AVL uses the aforementioned three files to assimilate all input data for analysis as shown in Figure 14. Upon program execution, trim conditions are determined, along with aerodynamic coefficients, control surface deflections, and stability axis derivatives about the CG. These are functions of C_L and angle of attack.

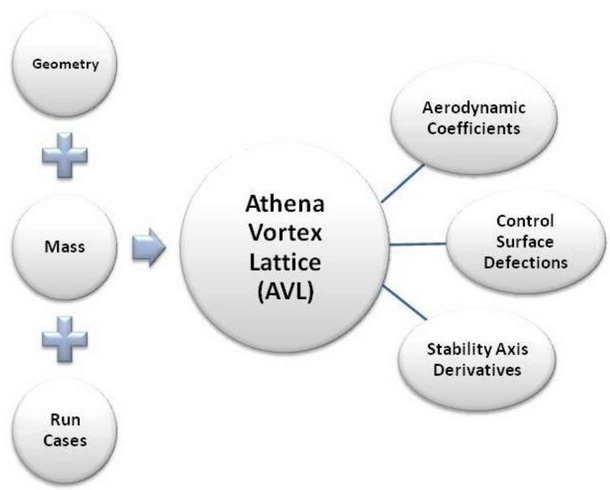

Figure 14. AVL inputs and outputs

Incorporation of AVL within the Model Center framework required several more files. AVL is a batch mode executable, so Analysis Server must be running in the background. Model Center recognized all the critical variables with a separate script called a “file wrapper.” The “file wrapper” tells Model Center exactly which files AVL needs, and where to locate all the important variables within the file. A “bat” file issues the required commands to the command prompt and a “batch” file spells out the sequence of operations to perform within the AVL context. Model Center requires a copy of each of these files to serve as templates. The original input file is modified to suit the Model Center user’s instructions while the “template” files are not changed between runs.

Lastly, there is an optional "initial" file used by Model Center to retrieve initial values. A copy of all of these files can be found in Appendix C.1.

III.2.1.3. AVL Operation: Outputs

AVL is a powerful tool capable of various complex analyses in its standalone form. However, when connected to Model Center, its power greatly increases by allowing the user to choose what is important. For example the conceptual design model uses only the span efficiency calculation from AVL for reasons mentioned later. This is not due to any shortfall, but it was determined to be the best fit with the other pieces of the model. Regardless of which pieces ultimately are connected to other components, by identifying variables in the "file wrapper," all output variables are accessible.

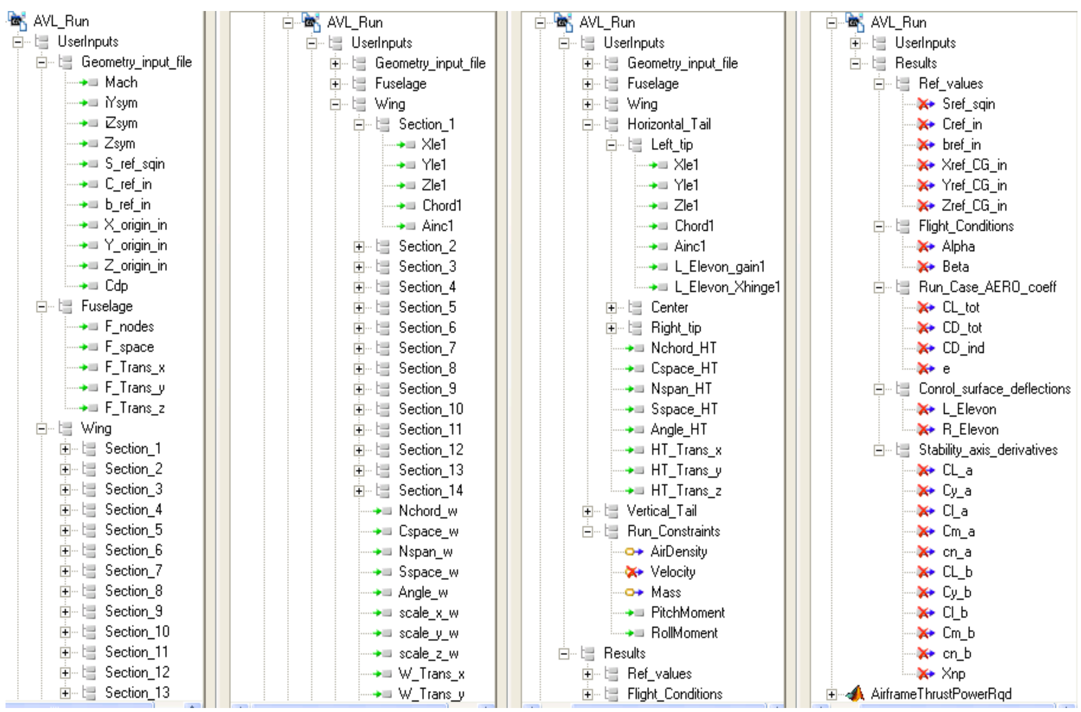

Figure 15. Partial breakout of AVL variables

A partial breakout of AVL input and output variables available from Model Center are displayed in Figure 15. In addition to standard aerodynamic variables, AVL

can also provide useful stability analysis. All of AVL's stability and control derivatives have been mapped to the Model Center interface. One simple illustration of a stability concern applicable at the conceptual design point is longitudinal stability. A positive static margin, identified by Eq. (15), suggests a longitudinally stable aircraft.[91]

$$\text{Positive Static Margin} = \overline{X}_{np} - \overline{X}_{cg} > \mathbf{0} \qquad \mathbf{(15)}$$

AVL provides a neutral point calculation, which is the aerodynamic center of total lift, each time it is run.

III.2.1.4. AVL Use in Model Center: ***Track 1*** and ***Track 2***

AVL outputs both span efficiency factor and total drag coefficient. The power-required and thrust-required can be calculated either from the drag polar constant, which is a function of span efficiency as shown in section III.2.4, or from the total drag coefficient. Within this conceptual design model, *Track 1* is denoted to be the airframe power-required calculation that uses drag polar coefficient, K, attained by including AVL's span efficiency, e. *Track 2* is denoted to be the airframe power-required calculation that uses AVL's total drag coefficient, C_D. Both of these calculation methods should produce identical results, and in fact only differ very slightly at slowest and fastest velocities. A comparison is provided in section IV.1.1. Because of these results, and the ease of attaining the span efficiency and drag polar constant, *Track 1* is used for all subsequent analysis, and *Track 2* has been relegated to a means of verifying *Track 1*.

III.2.2. Propulsion and Motor Model: QPROP

QPROP seeks to quantify the performance of propeller and motor combinations. Theory of operation is founded on classical blade element vortex theory and a detailed explanation is available through the QPROP website.[82] Circulation is defined as a

function of local chord length, velocity, rotation speed, and local radius. Parametric derivatives are determined and residuals are set to zero. A quadratic Newton iteration scheme is used to determine circulation for the flow condition. The circulation is then used to determine thrust and torque. By these means, an analysis can be performed based on velocity, thrust or torque, rotation rate, and pitch.

Inputs into QPROP cannot describe more than three design variables (velocity, thrust or torque, rotation rate, and pitch), or the problem will be over defined. When this occurs, QPROP tries to satisfy the inputs, but at least one output will not reflect the desired input. QPROP evaluates performance with Eqs. (16) - (25),

$$\eta_m = \frac{P_{shaft}}{P_{elec}} \tag{16}$$

$$\eta_{prop} = \frac{P_{prop}}{P_{shaft}} = J\frac{C_T}{C_P} \tag{17}$$

$$\eta_{overall} = \eta_m \eta_{prop} \tag{18}$$

$$P_{shaft} = Q\varpi \tag{19}$$

$$\varpi = RPM \cdot \frac{\pi}{30}, \text{[Hz]} \tag{20}$$

$$P_{prop} = VT \tag{21}$$

$$P_{elec} = Voltage \cdot Current \tag{22}$$

$$C_T = \frac{T}{(1/2)\rho(\varpi R)^2 \pi R^2} \tag{23}$$

$$C_P = \frac{Q}{(1/2)\rho(\varpi R)^2 \pi R^3} \tag{24}$$

$$J = \frac{V}{\varpi R} \tag{25}$$

where motor, propeller, and overall efficiency is η, shaft power, electric power, and propeller power is *P*, torque is *Q* , rotational speed is ω, velocity is *V*, thrust is *T*, and advance ratio is *J* with radius *R*.

QPROP can support any kind of motor, so long as a motor model file is developed. The basic configuration is for a brushed DC motor and uses the following Eqs. (26) - (29),

$$Q = \frac{(I - I_0)}{K_V} \tag{26}$$

$$\varpi = (V - IR)K_V \tag{27}$$

$$P = \varpi Q = (V - IR)(I - I_0) \tag{28}$$

$$\eta = \frac{P}{IV} = \left(1 - I\frac{R}{V}\right)\left(1 - \frac{I_0}{I}\right) \tag{29}$$

where *Q* is torque, *I* is current, *Io* is no-load current, K_v is in units of rad/s /Volt, ω is rotation speed, *V* is voltage, *R* is resistance in ohms, *η* is efficiency, and *P* is mechanical power in Watts. Ref. 82 provides further explanation of how these equations are used.

Outside the algorithm, the motor constant, K_v, is specified in traditional units of RPM/Volt. Resistance can include system resistance, which would be added to the motor resistance. A gearbox can also be included within the motor model. QPROP provides an optional motor model template that includes frictional torque, temperature-dependent resistance, and magnetic lags.

Unit conversion between English to Metric units is specified in the propeller template file. The QPROP algorithm requires SI units. All powers, efficiencies, advance ratio, thrust, and power coefficients are calculated values from inputs current, voltage, velocity, and RPM. The theory document explains the four parameters that are optimized: RPM, pitch rate change, thrust, and velocity. QPROP requires at least two of the four inputs, or it will not run. When more than three inputs are provided, the problem is over-constrained, and does not produce desired results. (When RPM is an open design variable, the desired thrust is returned). Figure 16 lists a partial breakout of QPROP variables. All QPROP program files can be found in Appendix C.5.

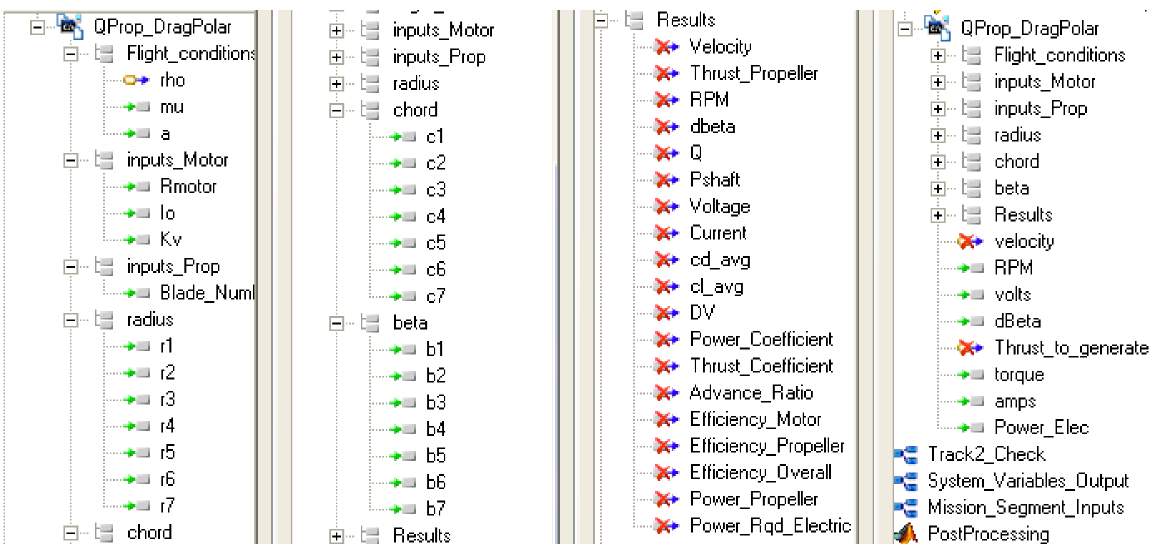

Figure 16. Partial breakout of QPROP variables.

III.2.3. Endurance Velocity, Cruise Velocity, and Rate of Climb

Best theoretical speeds for optimum performance relate directly to certain C_L/C_D ratios. The ideal ratio depends on whether the airplane is propeller driven or jet propelled. They can also depend on whether the propeller driven aircraft has a piston engine or an electric motor.

III.2.3.1. Endurance and Loiter Velocity

Endurance is how long an aircraft can stay in the air and is the same as loiter. Maximum endurance occurs when the aircraft flies at minimum power-required. A propeller driven aircraft achieves the maximum endurance when operating at minimum power. In order to operate at minimum power, it can be shown that the aircraft must fly at a ratio of $\left(\frac{C_L^{3/2}}{C_D}\right)_{MAX}$. Eq. (30) prescribes the velocity when this occurs.[36,91]

$$V_{end} = \left(\frac{2}{\rho}\sqrt{\frac{K}{3C_{Do}}}\frac{W}{S}\right)^{1/2} = \left(\frac{1}{3}\right)^{1/4} V_{(L/D)_{\max}} = 0.76 \cdot V_{(L/D)_{\max}} \qquad \textbf{(30)}$$

Equation (30) requires the drag polar constant, *K*, in order to provide the endurance velocity. The drag polar constant is dependent on span efficiency factor, *e*, which depends on the induced drag and lift coefficients shown in Eq. (8). Both the drag and lift coefficients are functions of velocity. Therefore, the drag polar constant indirectly depends upon velocity. Fortunately, this complicated relationship can be simulated with Model Center's converger application that will change the inputs of a function until the outputs match the inputs. In this case, a velocity is input into AVL. A drag polar constant is calculated from AVL's resulting span efficiency factor. The drag polar constant and input velocity is evaluated with Eq. (30) to find the endurance velocity. When the left hand side matches the right hand side, the equation has converged. This normally takes only a couple of iterations. This technique provides the best theoretical endurance velocity of V_{end} = 10.22 m/s for the GenMAV.

III.2.3.2. Range and Cruise Velocity

In much the same way as the endurance velocity corresponds to flight at the minimum power-required value, range is maximized by flying at the minimum airframe thrust-required. The cruise velocity is equivalent to the maximum range velocity. It can be shown that for a propeller driven aircraft, the maximum range velocity is the flight velocity of maximum lift-over-drag.[83] This formula is Eq. (31),

$$V_{(L/D)_{\max}} = \left(\frac{2}{\rho} \sqrt{\frac{K}{C_{Do}}} \frac{W}{S} \right)^{1/2} \tag{31}$$

which should not be confused with the minimum thrust-required and maximum range for a jet engine, which occurs at $\left(\frac{C_L^{1/2}}{C_D} \right)_{MAX} = (3)^{1/4} V_{(L/D)_{\max}}$.

Equation (31) also requires the drag polar constant, *K*, in order to provide the max range velocity. With the same relationship explained for the endurance velocity, *K* is dependent on the input velocity. Model Center's converger application was again used to change the input velocity and check the calculated range velocity. This technique provides the best theoretical best range velocity of V_{range} = 13.46 m/s.

III.2.3.3. Maximum Climb Rate Velocity

The maximum rate of climb is proportional to the excess power available, where excess power is the difference between power-available from the power plant at the flight condition and power-required to overcome drag while flying at the flight condition. As previously explained, the minimum power-required condition occurs during flight at the endurance velocity. The power-available, however, is strongly dependent on the type of power plant. A jet engine can produce increasing power with proportional increasing

velocity. Power-available from a propeller driven piston engine is relatively constant as velocity increases. Power-available from an electric power plant is strongly dependent on properly matching motor capability and propeller requirements, which infers that power-available will not be constant as velocity increases.[84] A poor match provides substantially low efficiency values. Furthermore, electric motor and propeller matching must be tailored to the design flight envelope because no one set of components can provide optimized performance for all conditions. QPROP can determine the operating efficiencies at any desired flight condition in order to calculate the power-required, but it cannot provide the power-available. This must be done experimentally. The real power-available curve for an electric powered MAV is likely to be as shown in Figure 17,[84] but without experimental data, the assumption is made that the most excess power will occur at a speed slightly higher than the endurance velocity. As such, the true maximum rate of climb cannot yet be determined.

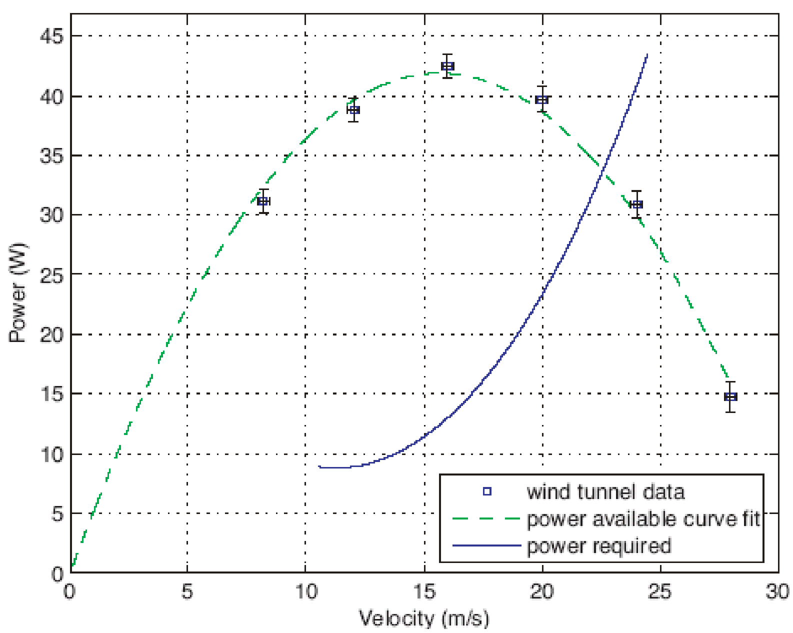

Figure 17. **P_A and P_R for small electric UAS.**

III.2.4. Airframe Power and Thrust Required

This section relies on the Equations of Motion (EOM) to develop theory necessary to determine the required thrust and required power for the airframe. The analysis is applicable for an electrically powered, propeller driven aircraft, and is independent of the inefficiencies associated with the motor and propeller.[83]

Beginning with Newton's Second Law, $F = ma$, and utilizing the Flat Earth frame of reference, the EOM for flight parallel to the flight path can be shown in Eq. (32),

$$m\frac{dV}{dt} = T\cos\varepsilon - D - W\sin\theta \tag{32}$$

where ε is the angle of thrust relative to the flight path, W is weight, V is velocity, D is drag, and θ is the instantaneous climb angle. The EOM for flight perpendicular to flight path in the vertical plane can be shown to be Eq. (33),

$$m\frac{V^2}{r_1} = L\cos\phi + T\sin\varepsilon\cos\phi - W\cos\theta \tag{33}$$

where ϕ is the roll angle and r_1 is the local radius of curvature of the flight path with respect to the vertical plane. A third EOM exists to describe motion perpendicular to the flight path in the horizontal plane, but two-dimensional motion is acceptable for a first-order conceptual design tool. Since we are only concerned with steady, unaccelerated flight, $\frac{dV}{dt} = 0$ and $\frac{V^2}{r_1} = 0$. Furthermore, a reasonable approximation can be made that the thrust line is in the direction of flight so that ε also = 0, so that Eq. (32) and Eq. (33) simplify to Eq. (34) and Eq. (35).

$$0 = T - D - W\sin\theta \tag{34}$$

$$0 = L - W\cos\theta \tag{35}$$

Lift is defined by Eq. (36), and drag is defined by Eq. (37), where the dynamic pressure, q , is defined by Eq. (38).

$$L \equiv qC_L S \tag{36}$$

$$D \equiv qC_D S \tag{37}$$

$$q \equiv \frac{1}{2}\rho V^2 \tag{38}$$

The coefficient of drag follows a pattern called the drag polar defined by Eq. (39), where $C_{D,o}$ is the zero-lift-drag coefficient and *K* is defined by Eq. (40).

$$C_D = C_{D,0} + KC_L{}^2 \tag{39}$$

$$K \equiv \frac{1}{\pi e AR} \tag{40}$$

A recent paper discussing the Unicorn MAV, which is of similar magnitude to the GenMAV, confirmed that this simple form of the drag polar is sufficient.[84] The drag polar constant Eq. (40) includes *e,* the span efficiency factor, and *AR,* the aspect ratio. Lift from Eq. (35) is substituted into Eq. (36), and the resulting lift coefficient is substituted into Eq. (39). A more elaborate form of Eq. (37) is then represented by Eq. (41).

$$D = qSC_{D,o} + \frac{KW^2\cos^2\theta}{qS} \tag{41}$$

The power-available, P_A, is primarily dependent on the ability of the power plant to produce thrust-available, T_A, and is the thrust multiplied by the velocity as shown in Eq. (42). The power-required, P_R, is the power the airframe requires to overcome drag and is shown in Eq. (43).

$$P_A = T_A V \tag{42}$$

$$P_R = DV \tag{43}$$

The Rate of Climb, *ROC*, is simply the trigonometric relationship between the vertical velocity verses the horizontal velocity represented by Eq. (44).

$$ROC = V \sin\theta \tag{44}$$

Conveniently, when the EOM from Eq. (34) is multiplied by V/W, a new EOM is expressed that includes the *ROC*, as shown in Eq. (45).

$$ROC = V \sin\theta = \frac{T_A V - DV}{W} = \frac{P_A - P_R}{W} \tag{45}$$

The drag from Eq. (41) can be inserted into Eq. (45) to yield Eq. (46).

$$ROC = \frac{T_A V - \left(qSC_{Do} + \dfrac{KW^2 \cos^2\theta}{qS} \right) V}{W} \tag{46}$$

Noting that airplanes typically fly at low climb angles, let $\cos\theta = 1$ for the drag expression only, so that Eq. (46) is simplified to Eq. (47).

$$ROC = V \left[\frac{T_A}{W} - \frac{1}{2} \rho V^2 \left(\frac{W}{S} \right)^{-1} C_{Do} - \frac{W}{S} \frac{2K}{\rho V^2} \right] \tag{47}$$

Ref 83 further asserts that this approximation has also been confirmed to be valid by W. Austyn Mair and David L. Birdsall, *Aircraft Performance*, Cambridge University Press, Cambridge, England, 1992. When the thrust-available is not known, as in the current effort, and the aircraft is desired to fly at a certain velocity and/or rate of climb, Eq. (47) can be solved for thrust, which then becomes thrust-required as shown in Eq. (48).

$$T_R = W\left[\frac{ROC}{V} + \frac{1}{2}\rho V^2\left(\frac{W}{S}\right)^{-1} C_{Do} + \frac{W}{S}\frac{2K}{\rho V^2}\right] \quad \textbf{(48)}$$

Eq. (48) is very useful because it is valid for both climbing and non-climbing flight by setting $ROC = 0$ for level flight.

Models that represent the airframe, propeller, motor, and power required have been described in this section. The next section discusses the power system models.

III.2.5. Fuel Cell Model

Given that fuel cells are highly multi-disciplinary, the literature contains a broad range of potential fuel cell models for various applications. Perhaps the most striking difference among the models is the frequency of which scientists and engineers from different backgrounds produce models tailored to a single discipline while minimizing or ignoring the rest. In this case, only a high-level mechanical engineering perspective is required, while the details of thermodynamics, and electro-chemical interactions are less of a concern. The following model produces the polarization curve.[75]

Theoretical open circuit voltage for a hydrogen fuel cell is given by $E = \frac{-\Delta\overline{g_f}}{2F}$, where E is the reversible electromotive force (EMF) or reversible open circuit voltage, $\Delta\overline{g_f}$ is the molar specific Gibbs free energy, and F is the Faraday constant. Gibbs free

energy ([kJ/mol]) varies with temperature. The product will be water below 80 °C and vapor above this temperature.

Loss models are specific to the type of loss. Activation losses can be estimated by the Tafel or Butler-Vollmer equation, given by Eq. (49) and Eq. (50),

$$\Delta V_{activation} = A \ln \frac{i + i_n}{i_0} \quad \textbf{(49)}$$

$$A = \frac{RT}{2\alpha F} \quad \textbf{(50)}$$

which includes crossover and internal current density losses. The charge transfer coefficient, α, is dependent on type of reaction and electrode material. It can be approximated as 0.5. The universal gas constant, *R*, is 8314 J/mol. Temperature, *T*, is in Kelvin. Current density, *i*, has units of mA/cm^2. Exchange current density, i_o, is dependent on electrode material, and ranges from 2.5x10^{-13} to 4x10^{-3} A/cm^2. The internal current density, i_n, is often negligible. Ohmic losses due to electrical resistance of electrodes and ion flow follow Eq. (51),

$$\Delta V_{ohmic} = i \cdot r \quad \textbf{(51)}$$

where *r* is the area specific resistance in kΩ/cm^2. Mass transport loss empirical equation is shown as Eq. (52),

$$\Delta V_{transport} = m \cdot e^{(n \cdot i)} \quad \textbf{(52)}$$

where *m* and *n* are constants of approximate values of 3x10^{-5} volts and 8x10^{-3} cm^2/mA. Together, all losses of Eqs. (49), (51), and (52) combine into the Eq. (53).

$$V = E - i \cdot r - A \ln \frac{i + i_n}{i_0} + m \cdot e^{(n \cdot i)} \quad \textbf{(53)}$$

Eq. (53) can be reduced as Eq. (54) for most cases.

$$V = E_{oc} - i \cdot r - A \ln(i) + m \cdot e^{(n \cdot i)} \quad \textbf{(54)}$$

Hydrogen storage requirements can be found from Eq. (55), and hydrogen flow rate can be determined from Eq. (56),[50] but under predicts the flow rate, possibly due to leaks and hydrogen crossover effects.

$$n_{H_2} = \frac{n_{cells} IE}{2F\eta_{util}} \quad \textbf{(55)}$$

$$\dot{V}_{H_2} = \frac{MW_{H_2}}{\rho_{H_2}} \frac{n_{cells} I}{2F\eta_{util}} \quad \textbf{(56)}$$

Hydrogen utilization, η_{util}, is approximately 90%. The Faraday's number, *F*, is 96485 C/mol. The cell current is *I*, and *E* is the aircraft endurance. The number of cells is n_{cells}.

The polarization equation shown in Eq. (54) was originally built into this conceptual framework. Later, it was decided that a more practical first step would be to rely solely on specific energy and specific power values, thereby only symbolically including a fuel cell into the framework. Utilizing an actual fuel cell model as described here could provide more conclusive results regarding fuel cell hybridization.

III.2.6. Battery Model

In the same context as the fuel cell model, two battery models were originally included within the conceptual design framework. Later, it became evident that a more useful tool could be built within the available time that utilized specific power and specific energy as proxies for a more accurate battery model. This is especially true since

the State of Charge (SOC) is not tracked and battery charging capability is not developed here. However, the simple models presented here would be an ideal next step.

As shown in Figure 3 and Figure 4, battery cell voltage depends on the SOC and discharge rate. A higher voltage is available at a lower discharge rate and careful monitoring will help ensure terminal voltage stays within the middle linear region. A standard battery can be modeled as a current source in series with an internal resistance as shown in Figure 18.[77] Terminal voltage follows Eq. (57) where the current source voltage is E, terminal voltage is V, and internal resistance is R_0.

$$V = E - IR_0 \qquad \textbf{(57)}$$

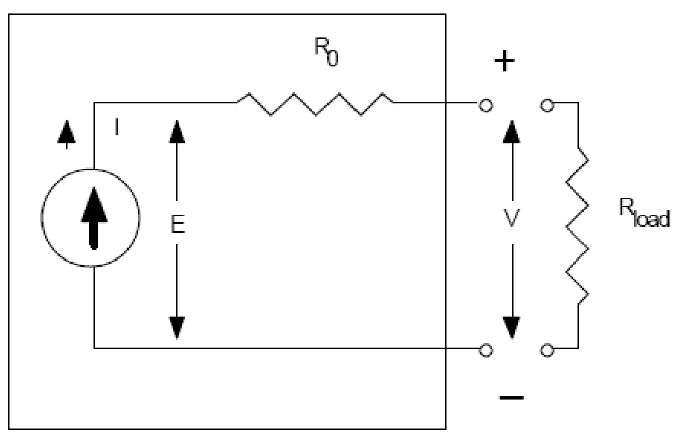

Figure 18. Standard battery model schematic.

A very common type of battery cell is the 18650 lithium polymer, which was used in AeroVironment's 150 g, 30 cm WASP MAV.[24] It supplies 3.7 volts per cell and is connected in series for higher voltage. The power supplied by battery cells connected in series can be modeled by Eq. (58).[4]

$$P_{batt} = V_{oc} I_b{}^2 R_{\text{int}} \tag{58}$$

A slightly more complicated model is offered by reference 78. This equivalent circuit is shown in Figure 19 and follows Eq. (59).

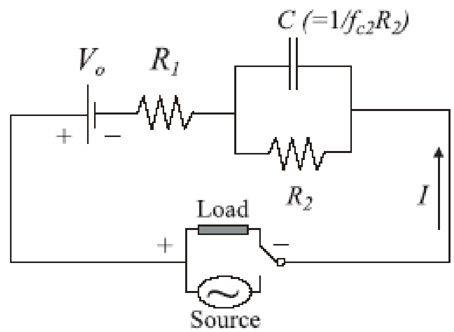

Figure 19. Equivalent circuit for lithium ion cells.

$$V(t) = \frac{Q}{C} e^{-t/\ R_2 C} + V_0 - IR_1 - IR_2 (1 - e^{-t/\ R_2 C}) \tag{59}$$

It is well suited for lithium ion cells. The current, I, is constant. The nominal capacity is Q, and V_0 is the nominal SOC dependent open current voltage.

Individual fuel cell and battery models need to be integrated to provide system level power. Another model called the power management model brings the power subcomponents together.

III.2.7. Power Management Model

Fuel cells lack the required robustness to power a system with highly fluctuating power needs, as is theorized for future MAVs. Therefore, in order to build or retrofit a useful MAV with today's COTS materials, the most likely power scheme is a fuel cell-battery hybrid system.

Fuel cells provide high specific energy and batteries contribute high specific power. These dominant attributes dictate the sizing of each component. In order to

capitalize on the fuel cell's main benefit, the fuel cell must run at its most efficient operating point. A fuel cell runs most efficiently when its load is constant and when that load is relatively low. Therefore, the accepted convention is to size the fuel cell for average load throughout the mission according to Eq. (60).

$$P_{Bat} = P_{tot} - P_{FC(avg)} \qquad \textbf{(60)}$$

It was previously mentioned that in the interest of providing a tool for the power system designer to use in the very near future, a SOC battery model and a polarization curve dependent fuel cell model are not yet incorporated into the conceptual design tool. Therefore, a first step is to base the battery and fuel cell on their specific energy and specific power properties. Eq. (60) is now recast into Eq. (61) using the terms of Table 5,

Table 5. Specific power and specific energy notation.

Energy Dense Source, Specific Power	(EDS_{SP})
Energy Dense Source, Specific Energy	(EDS_{SE})
Power Dense Source, Specific Power	(PDS_{SP})
Power Dense Source, Specific Energy	(PDS_{SE})

$$P_{PDS} = P_{total} - P_{EDS(avg)} \qquad \textbf{(61)}$$

where the fuel cell represents the Energy Dense Source (EDS), and the battery represents the Power Dense Source (PDS).

The battery, capable of providing higher short-term power, is sized to level the peaks of the power profile. This may not necessarily be optimal. The following relations lay the foundation to find the answer to this question on a case-by-case basis. The conceptual design tool includes a power profile generator that provides a vector of power values based on the flight conditions of each segment. Each mission segment has a

distinct duration and represents average power except for climb and descent where some post processing of multiple measurements reduce to a segment power average. The mission power average can then be determined by Eq. (62).

$$P_{average} = \frac{\sum \left(t_{segment} * P_{segment\_average} \right)}{\sum t_{total}} \quad \textbf{(62)}$$

$$P_{EDS} = \Delta * P_{average} \quad \textbf{(63)}$$

EDS power is determined with Eq. (63) by a user input denoted as *differential* and shown with the symbol Δ. The differential is set as a certain percentage of the average power. Time is represented by *t*. A value of zero calls for no EDS, where a value of 1 or 100% sets the EDS power to be the average mission power, a typical arrangement. The PDS then provides the additional power per Eq. (64).

$$P_{PDS} = P_{\max} - P_{EDS} \quad \textbf{(64)}$$

Energy, *E*, is simply the power times the time shown in Eq. (65). Energy requirements for the PDS accumulate across the mission segments according to the PDS power requirement shown in Eq. (66).

$$E_{EDS} = P_{EDS} * t_{total} \quad \textbf{(65)}$$

$$E_{PDS} = \sum_{0}^{Max\_Segment\_\#} \left((P_{segment} - P_{EDS}) * t_{segment} \right) \quad \textbf{(66)}$$

The PDS and EDS must be sized to meet both power and energy requirements. In nearly all cases, the source will have either excess power or excess energy. The power is an instantaneous requirement, and the energy relates to the endurance of the aircraft. Often, excess energy must be carried as a consequence of meeting the power

requirement. In order to address both requirements, the mass of each component was calculated based on the ratio of the requirement and the specific power or energy property, as shown in Eqs. (67) - (71).

$$m_{PDS\_(energy\_driven)} = \frac{E_{PDS\_required}}{E_{PDSSE}} \quad \textbf{(67)}$$

$$m_{PDS\_(power\_driven)} = \frac{P_{PDS\_required}}{P_{PDSSP}} \quad \textbf{(68)}$$

$$m_{EDS\_(energy\_driven)} = \frac{E_{EDS\_required}}{E_{EDSSE}} \quad \textbf{(69)}$$

$$m_{EDS\_(power\_driven)} = \frac{P_{EDS\_required}}{P_{EDSSP}} \quad \textbf{(70)}$$

Both the energy requirement and the power requirement of each component determined a minimum weight. In order to satisfy both requirements, the largest of the power and energy mass is then selected for each component as shown in Eqs. (71) and (72).

$$m_{PDS} = \max\left(m_{PDS\_(power\_driven)}, m_{PDS\_(energy\_driven)}\right) \quad \textbf{(71)}$$

$$m_{EDS} = \max\left(m_{EDS\_(power\_driven)}, m_{EDS\_(energy\_driven)}\right) \quad \textbf{(72)}$$

Total power system mass is the sum of the PDS and EDS mass subject to an overhead mass percentage penalty for power system packaging shown in Eq. (73). In the model, 10% was used for packaging.

$$m_{PowerSystemTotal} = overhead * \left(m_{PDS} + m_{EDS}\right) \quad \textbf{(73)}$$

This power system model would be just as appropriate for comparing two batteries with different properties as it is for comparing a battery (PDS) and a fuel cell

(EDS). A useful battery-battery hybrid-power source would have one with high specific energy and another with a high specific power.

As suggested elsewhere in this text, it is important for the user to understand that maximum specific energy and maximum specific power is not available simultaneously. Therefore, it is imperative that the user recognize and input reasonable values for reasonable results.

Each component described in this chapter, such as AVL, QPROP, and power management can be integrated within the Model Center environment. Model Center provides many means to run models and better understand data.

III.3. Model Center Trade Studies

Model Center offers a variety of tools useful for analysis. Of the simplest is the parametric study, which captures output variable values as input variable parameters are changed. A bit more complex is the Design of Experiments (DOE), which helps it identify the most important design variables. Most complex are the optimizations. Not only can a model be set to determine maximums, minimums, or find specific values, a whole suite of data visualization tools help interpret the best designs.

III.3.1. Parametric Study and Carpet Plots

A parametric study in Model Center is a sweep on an input variable while capturing results of a single or any number of output variables. This is useful for identifying trends and locating candidate design points for optimization. More complexity is added when a carpet plot is run. A carpet plot is a two-dimensional parametric study. The number of steps and step size are set by the user. After the carpet

plot is run, surface and contour plots are readily available. These are very useful in realizing dependencies and constraint effects not evident with the parametric plot.

III.3.2. Design of Experiments

Often, with complicated models, the magnitude of effects caused by changing a certain input variable is unclear. Laboratory experimentation across the spectrum of possible input conditions is usually too costly and time consuming. DOE tools simplify this problem. By modifying potentially important input variables and then tracking the output effects, a well designed DOE can statistically illustrate which variable are most important or which conditions are most sensitive to change. Model Center allows the user to select one of twelve algorithms to correlate with most appropriate desired DOE application. The default is a full-factorial, which may or may not be the best choice, depending on desired results. A standard output of Model Center's DOE tool is the "main effects" plot. This plot identifies the impact of each model variable on the response variables.

III.3.3. Design Optimization Tools

Model Center also offers the ability to perform optimization. The Design Optimization Tools (DOTs) modify the design variables until a specified condition is met. These are cast in the conventional nomenclature by using an objective function, constraints, and convergence criteria. Instead of a mathematical cost function, optimization is performed by analyzing the results from running the model's components. Model Center's DOT methods include variable metric, conjugate gradient, feasible directions, sequential linear programming, and sequential quadratic programming.

Various outputs such as convergence plots and iteration reports are available after the optimization completes.

III.4. Validation Platform

Given that various MAV platforms are not easy comparable in this nascent research field, the AFRL Munitions Directorate decided to create a platform that will be used as a baseline for future collaboration.[39,61] This platform is called GenMAV, and will be modeled in this study as well. The GenMAV has a positive 7° dihedral, 5° angle of incidence high wing, circular fuselage, tractor propeller, and empennage with a standard tail. The wingspan is 60.2 cm with a 12.7 cm chord length. AVL was used for GENMAV. All analysis thus far has been through AVL, flight tests are underway. The system power requirements could be validated by putting a small voltage and current recorder in the GenMAV to get the power. A second GenMAV with an articulated wing, called the GenMAV 2, was created to add flexibility and better study the aerodynamic control. The AFRL Munitions Directorate graciously shared the AVL model for this platform, and it therefore became the baseline platform for this study. Figure 20 is a picture of the GenMAV 2.[61]

Figure 20. Flight-Ready GenMAV 2.

Chapter 3 discussed all of the various subcomponents within the conceptual design tool. Each piece of software relied on individual models that were not initially meant to interoperate. The Model Center environment is built for that connectivity, which in turn can provide relatively simple systematic analysis capability. One must first test each component within the model before taking the system-view. To this end, Chapter 4 shows how each major model was verified within the Model Center framework. Then system-level parametric studies were performed to both confirm good operation and provide insight into the test article. Lastly, mission simulations and power system optimization provided useful results.

IV. Analysis and Results

IV.1. Verification and Validation (V&V)

This effort was as much about building a model as researching useful results. Before any actionable information can be attained from the conceptual design tool, each major sub-model must be verified and checked to confirm that it not only acts as designed, but also produces accurate results. All analysis was performed from the Model Center GUI. *Verification* confirms that something performs as designed. *Validation* confirms that something produces desired results. AVL and QPROP will be verified and validated as much as possible.

IV.1.1. AVL Verification

An important means to verify AVL is to iteratively determine the drag and lift coefficients, and plot the drag polar for visual inspection. A range of velocities at sea level was input into the AVL GenMAV model and vectors of lift and drag coefficients were collected. Preliminary wind tunnel tests on the GenMAV 2 show a zero-lift-drag coefficient of 0.102. This value was input into AVL, and used for all analysis. AVL failed to converge for velocities less than 9.23 m/s, when it produced lift coefficient requirements higher than the known maximum lift coefficient. Preliminary wind tunnel data also predicts a maximum lift coefficient of 1.16, which was used as the maximum lift coefficient within the model. Considering the power required to fly at high speeds, a reasonable upper bound is 25 m/s. Figure 21 shows the derived drag polar associated with the GenMAV for the given input conditions. The data are curve fit to a linear model shown by Eqs. (74) and (75).

$$C_D = C_{D_0} + KC_L{}^2 \tag{74}$$

$$C_D = 0.1038 + 0.0637C_L{}^2 \tag{75}$$

Graphically shown by Figure 21, the linear model matches the data set. The use of a single value for K is acceptable for a first-order approximation.[84] It can be seen from this assessment that AVL does a adequate job of honoring the prescribed value of the zero-lift-drag coefficient.

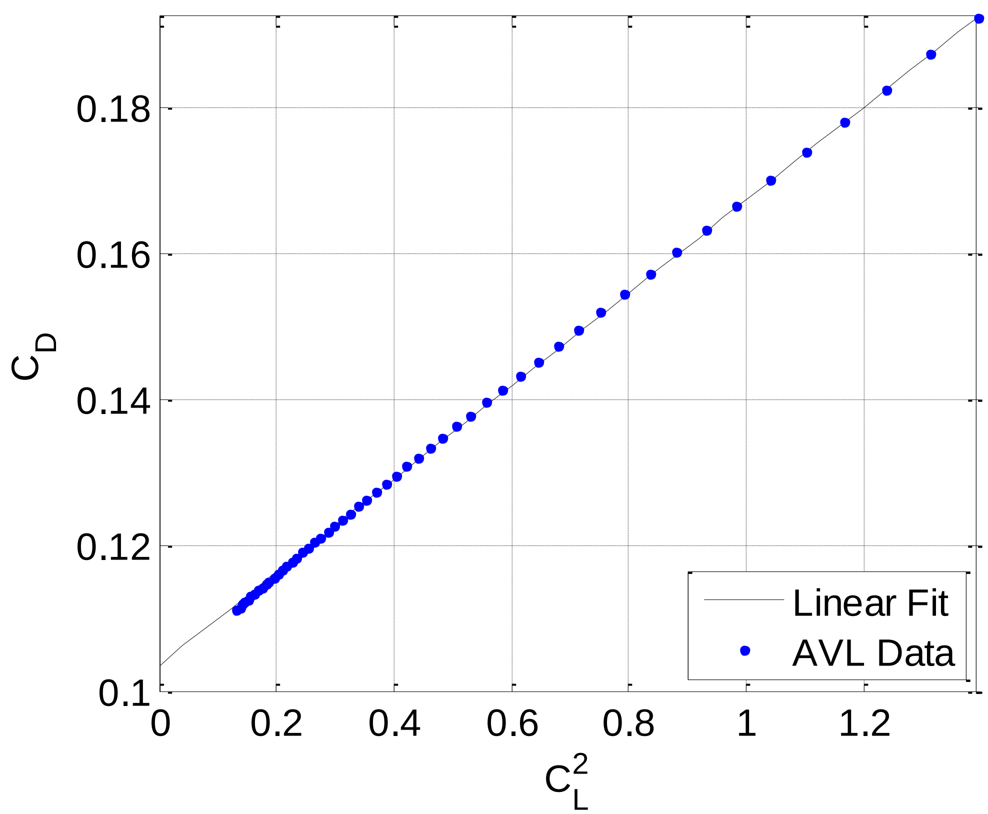

Figure 21. AVL generated GenMAV drag polar.

The drag polar shown in Eq. (75) was calculated for the GenMAV based on velocity runs with AVL. The resulting value of K within the drag polar is the slope of the line on the plot of C_D vs. C_L^2. Knowing the aspect ratio, AR, and the drag polar coefficient, K, the average span efficiency factor, e, is computed with Eq. (76).

$$K = \frac{1}{\pi e AR} = 0.0637, \quad e = 0.97 \tag{76}$$

Such a high span efficiency is likely to be optimistic. Actual span efficiencies range from approximately 80-95% and shall be illustrated in Figure 34. However, it is entirely possible that AVL provides span efficiency values not directly comparable to conventional values. As previously mentioned, the conceptual design tool has two somewhat distinct tracks. *Track 1* utilizes the span efficiency output of AVL to calculate a drag polar constant, which is then used to determine the endurance and maximum range velocities. *Track 2* is a means to check *Track 1* by using AVL's total drag coefficient to determine minimum power and minimum thrust. Figure 22 illustrates power and thrust results for both of these approaches. Both overlap, as they should, except at the lowest velocity. The difference is due to *Track 2* not incorporating a changing span efficiency. This appears to be a negligible difference. The minimum points within the data are displayed in Table 6.

Use of the converger to determine minimum velocities is considered to be closer to the true analytic value because the feedback loop confirms that the span efficiency and drag polar coefficient are matched to the input velocity. A comparison of the numbers of Table 6 identify *Track 1* as the better choice. Therefore, *Track 1* was used for all analysis, and *Track 2* was only retained as an additional check.

Table 6. Calculation of endurance and maximum range velocities.

#	Approach	Vend (m/s)	Vmax_range (m/s)
1	Model Center converger function	10.23	13.46
2	AVL Velocity Iteration (*e*), *Track 1*	10.47	13.32
3	AVL Velocity Iteration (C_D), *Track 2*	10.68	13.27

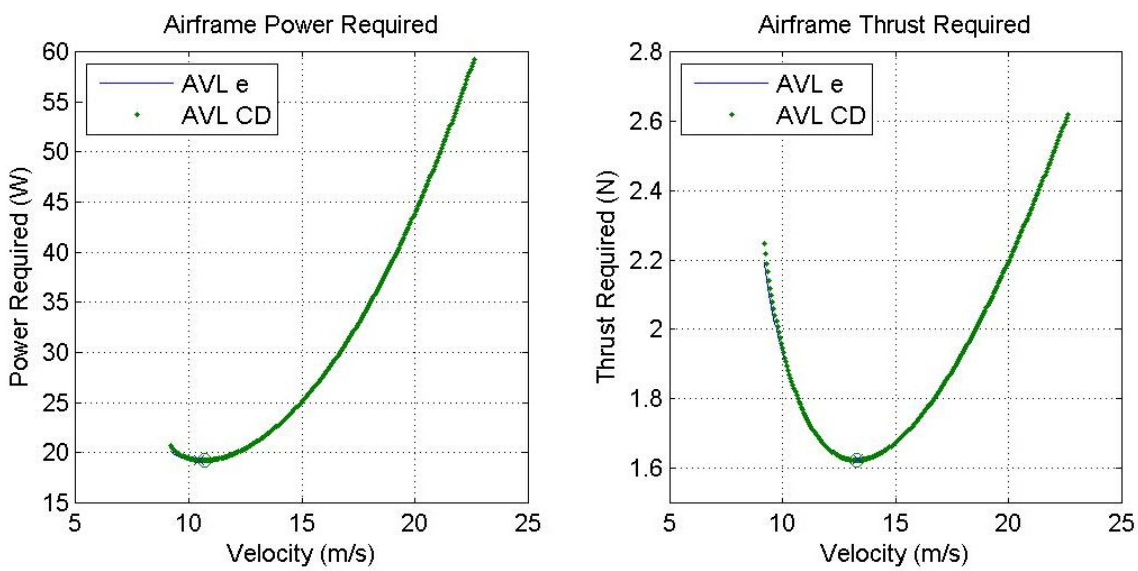

Figure 22. Iterative approach to find endurance and max range velocity.

IV.1.2. AVL Validation

Validation of AVL is not possible at this time since wind tunnel data was not obtained. However, an AFRL team is currently evaluating GenMAV wind tunnel results and will publish them soon. Preliminary results look very promising.

IV.1.3. QPROP Verification

QPROP predicts performance of propeller and motor combinations. Without deriving the theory of operations, a good method of verifying that the code works properly is to check the outputs with values obtained independently.

Figure 23 illustrates data directly from QPROP. Advance ratios, thrust coefficients, and power coefficients are direct outputs from QPROP and are available to the user without further processing. As a check to better understand the output, the QPROP Eqs. (16)-(25) were independently recalculated in MatLab using only non-derived variables such as velocity, power, thrust, and RPM. The independent calculations overlap QPROP outputs, so this proves that the QPROP outputs follow the provided equations.

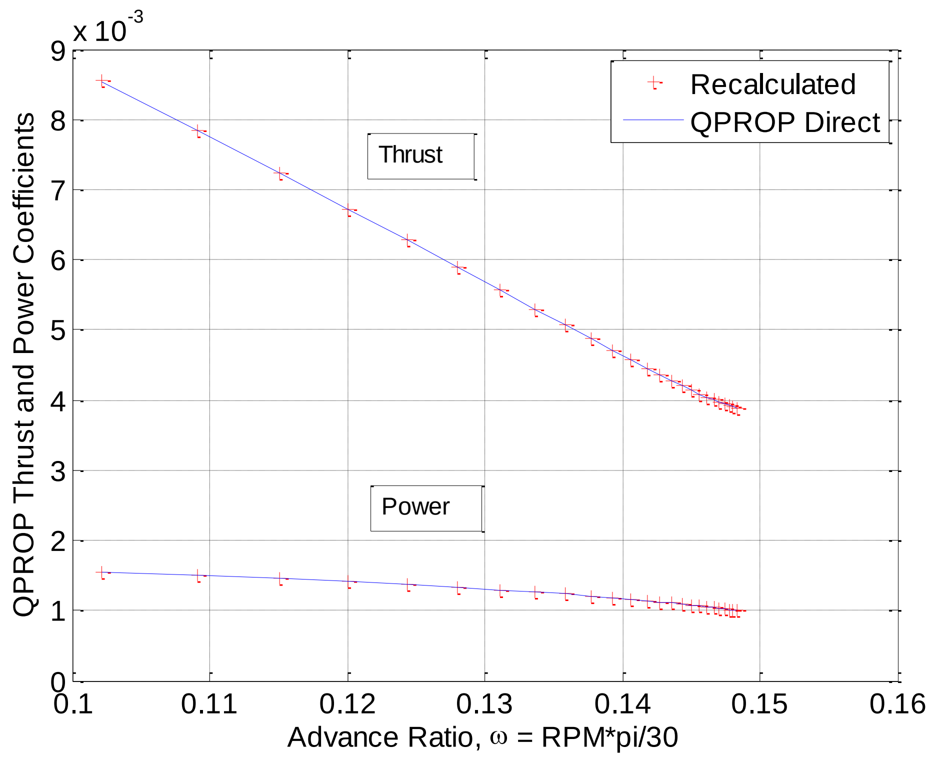

Figure 23. Verification of QPROP application of Eqs. (16)-(25).

Equation (17) described the propeller efficiency and is therefore very important to any model. This equation was checked with Figure 24 by calculating the efficiency using the underlying thrust and power coefficients. The calculated values overlap QPROP outputs perfectly.

One concern about QPROP outputs is that magnitudes of coefficients and advance ratios are much smaller than one would expect. After appropriately scaling the outputs, it was found that QPROP actually does produce expected results shown in Figure 25 through Figure 27. Further explanation is provided in subsequent pages. When utilizing individual QPROP outputs, it would be important for any user to confirm that Eqs. (16)-(25) reflect the user's desired methodology.

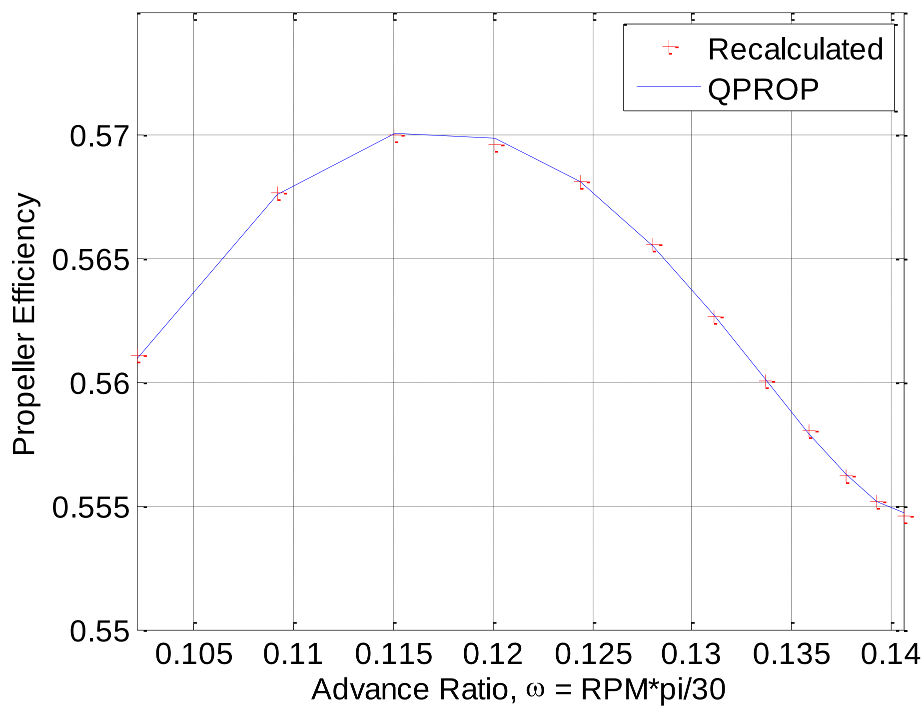

Figure 24. Verification of QPROP's application of Eq. (17).

IV.1.4. QPROP Validation

Validation posed a challenge due to the availability of data and nonstandard data reporting procedures. However, reference 93 provides test data for propellers similar to the one chosen for this work. The research presented in this reference has two important conclusions. First, prediction capability is poor at low Reynolds numbers. Second, scatter at low Reynolds number exists because advance ratios are defined in numerous ways.

Research presented here is based on a 9.5x5 Aeronaut (AE) CAM propeller. However, geometric data for for this propeller, as required by QPROP, was not available. Instead, available Graupner 6x3 propeller data was extrapolated up to the correct sizes. In hindsight, this may have not been wise since test data is not available for this imaginary propeller. Despite this, it still allowed for high level analysis, and can be improved in the future. Test data found in the above reference came from experiments performed at the Langley Aeronautical Research Center's Basic Aerodynamics Research Tunnel (BART). Comparable data are for two different 9.5x5 AE CAM folding propellers.

In order to compare data, it was first necessary to recompute advance ratios and associated coefficients into the same form presented in the test data. These equations are Eqs. (77)-(80),

$$C_T = \frac{T}{n^2 D^4} \tag{77}$$

$$C_Q = \frac{Q}{n^2 D^5} \tag{78}$$

$$n = \frac{RPM}{60} \tag{79}$$

$$J = \frac{V}{nD} \tag{80}$$

where C_Q is the torque coefficient, n is the propeller rotational speed in revolutions per second, and D is the propeller diameter.

Although not re-represented here, the BART data trends identically to the QPROP data shown in Figure 25 and Figure 26. The thrust coefficient slopes downward linearly at approximately the same pitch shown. The torque coefficient follows the same concave slope as shown. At the same advance ratios, the QPROP thrust coefficient results are approximately 30% less than the BART data. Also, at the same advance ratios, the QPROP torque coefficient results are approximately 35% less than the BART data.

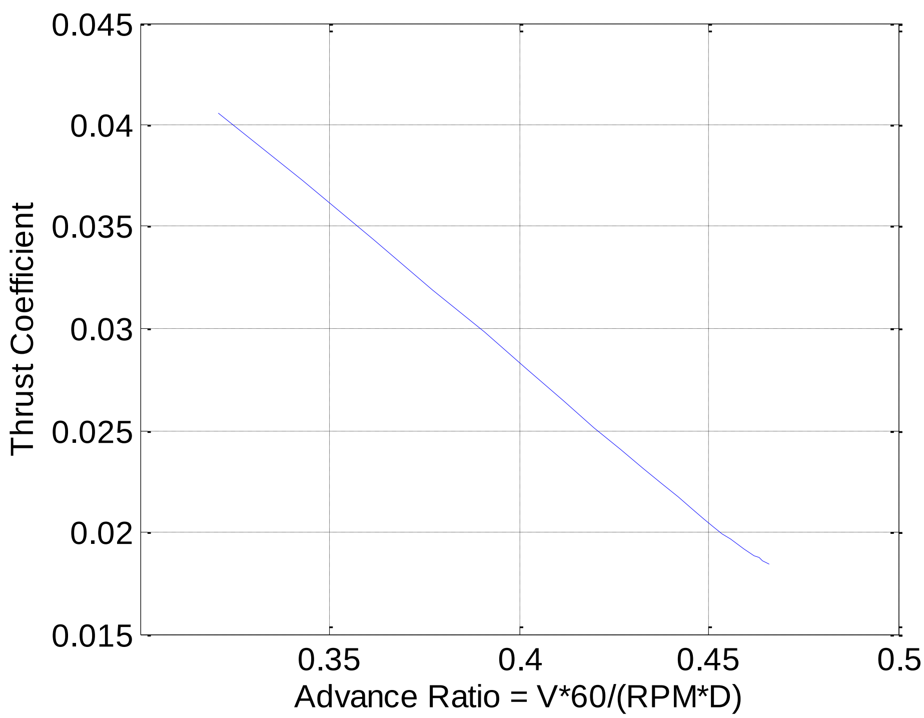

Figure 25. QPROP thrust coefficient data.

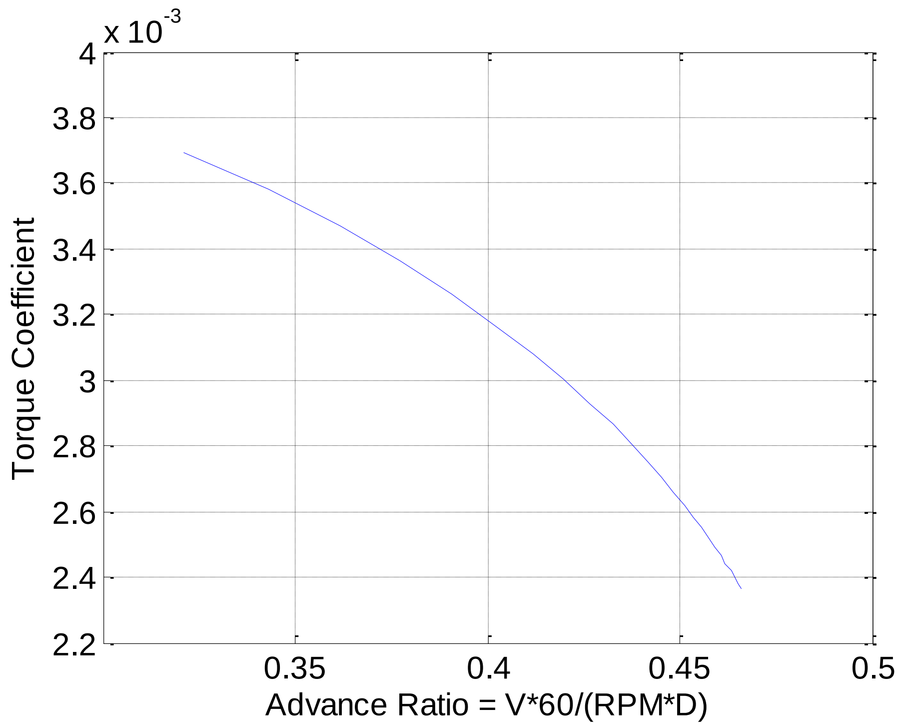

Figure 26. QPROP torque coefficient data.

BART test efficiency data were also compared to the "imaginary" propeller. For the selected test propeller, efficiency data exists at an advance ratio of 0.2 to 0.6. This range is comparable to QPROP advance ratio range of 0.33 to 0.43 as shown in Figure 27. The test data matches QPROP data in magnitude of approximate maximum efficiency of 57% at near the same advance ratio of 0.37. Another researcher found that for a single GWS-HD 9x5 propeller tested in the University of Illinois wind tunnel, QPROP over-predicted maximum efficiency by 17% within the same advance ratio range as the BART tests.[94] The QPROP under-prediction of thrust coefficients when compared to BART data and the QPROP over-prediction of efficiency when compared to a test performed on a different propeller at an Illinois wind tunnel may appear to be a contradiction, but it is not. Efficiency is determined by QPROP with Eq. (17), which

includes the ratio C_T/C_P. If both of these coefficients are calculated by similar methods by the same source, the ratio should not be different. Therefore, efficiency data may not be proportional to independent non-dimensionalized coefficients. Data available from Illinois is of interest because someone utilized QPROP, but a proper comparison to research presented here is not applicable because the propeller geometry was not rigorously modeled. Obviously, a better propeller geometry file could have been developed, but at this point, all that can be surmised is that QPROP provides adequate estimations for conceptual design.

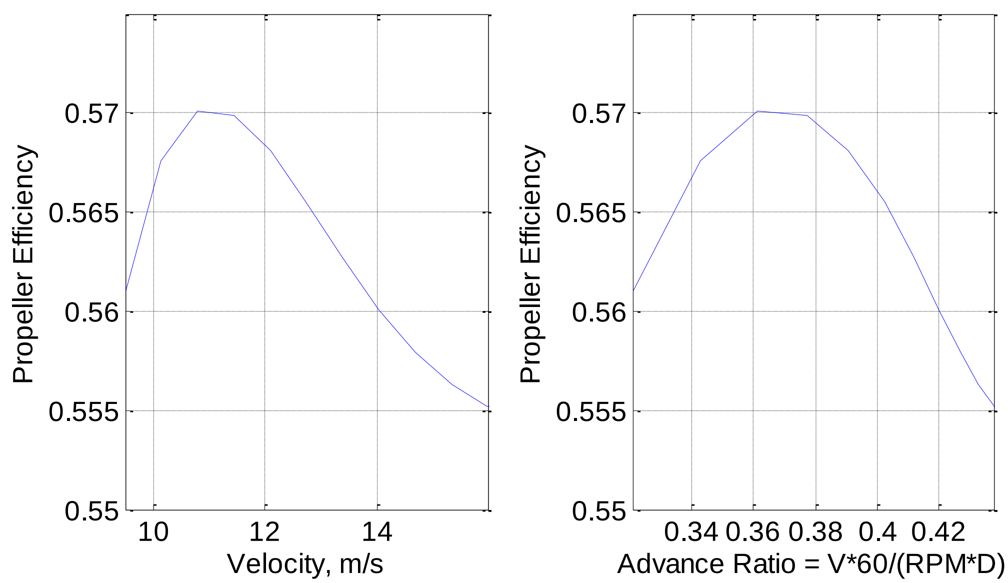

Figure 27. QPROP propeller efficiency data.

The GenMAV uses an AXI 2808/20 motor. This is a brushless DC outrunner with 0.105 ohms resistance, 1.30 A no-load current and a voltage constant of 1490 RPM/Volt. The motor model was not directly validated, but a comparison to a similar type of motor in the literature was used to provide a rough check on results. Most literature suggests that manufacture's data should not be trusted and test data was not

found for this specific motor. However, a research paper was found that specifically investigated motors of approximately the same size and operating conditions. This paper identifies a maximum efficiency of between 75 and 85%.[85] QPROP results show a maximum motor efficiency of 77%, which is within the acceptable literature range.

IV.1.5. Power Profile Verification

An example power profile will be shown in section IV.3. As expected, the maximum power-required for the given mission occurs when the vehicle is climbing. Power-required then increases with altitude. During the cruise portion of the flight, power is reduced some, but remains high due to flight faster than endurance speed. The loiter mission segment is flown at endurance speed, which has the minimum power-required. The descent, naturally, requires the least power. These results trend as expected.

IV.1.6. Power Profile Validation

Power profile validation will require flight testing. Lightweight inductance Ampere meters are now available for flight test data logging. Candidates for flight test vehicles, if not the GenMAV, are the BATMAV, the BATCAM, the AeroVironment Raven, or the AFIT OWL. A flight vehicle other than the GenMAV could still validate trends and power proportions between flight segments.

IV.2. Parametric and Carpet Plot Analysis

Model Center offers a wide variety of means to analyze data. Section III.3 introduced some of the trade study capability available once the complete model has been built. The various disciplines and software that make up this MAV conceptual design

tool must be comprehensively tested to ensure that they each provide the desired functionality. The best way to perform these tests is to run a series of parametric studies. After initial verification, it is not too difficult to modify the model to produce speculative simulations with far greater ease than by experiment. Later, powerful optimization can highlight immediate possible improvements.

In order to further verify the model and explore the GenMAV capability, a set of outputs were analyzed against three changing inputs: altitude, MAV mass, and rate of climb. For all cases, the input freestream velocity was increased in finite steps from 9.5 m/s to 25 m/s. Altitude was varied from 0 to 500 m, total MAV mass was varied by approximately +/- 25%, and rate of climb was varied between -1 to 1 m/s. The outputs were motor efficiency, propeller efficiency, overall efficiency, airframe power required, and total electric power required.

IV.2.1. Velocity and Altitude

When compared to velocity, altitude plays a minor role in the observed motor efficiency outputs as seen in Figure 28 and Figure 29. Motor efficiency increases to 77% percent as velocity increases. Given the slope, it could increase further. Altitude is shown to be more important at low speeds than at high speeds. Propeller efficiency on the other hand, appears generally unaffected by velocity until a relatively high speed when efficiency increases by a small amount as shown in Figure 30 and Figure 31. The range between 55 and 57% is typical. Given the increasing trend, propeller efficiency was calculated up to 50 m/s (not shown) to find a maximum of 64% at 41 m/s. This demonstrates that the propeller is not matched to the motor or the airframe, and the efficiencies given by Figure 27 and the slower region of Figure 30 are local maximums.

The models confirm that the propeller is not ideal for the GenMAV. The abilty of the conceptual design framework to illustrate this poor match further verfies the tool. Figure 30 also shows that for a given velocity, altitude impedes propeller efficiency, as expected. Overall, a higher speed is required to maintain efficiency at higher altitudes, as shown in Figure 32 and Figure 33.

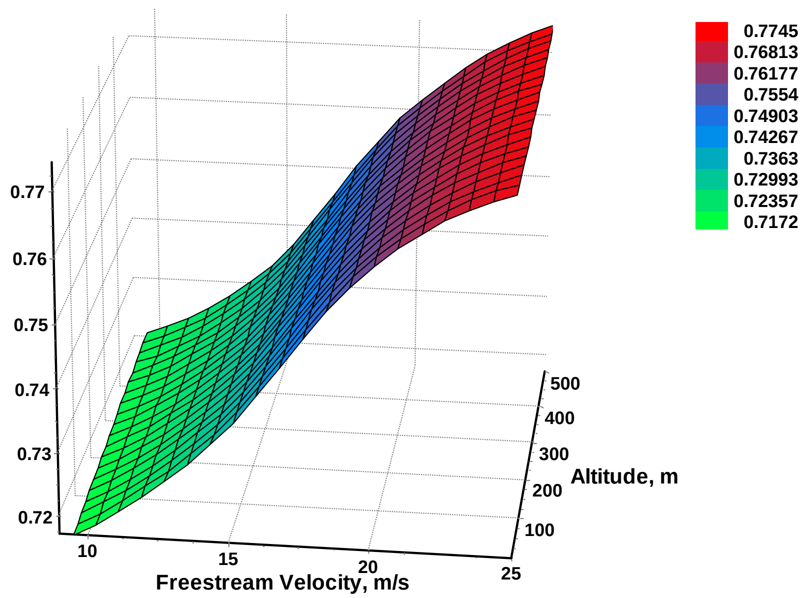

Figure 28. Motor efficiency vs. velocity and altitude, carpet plot.

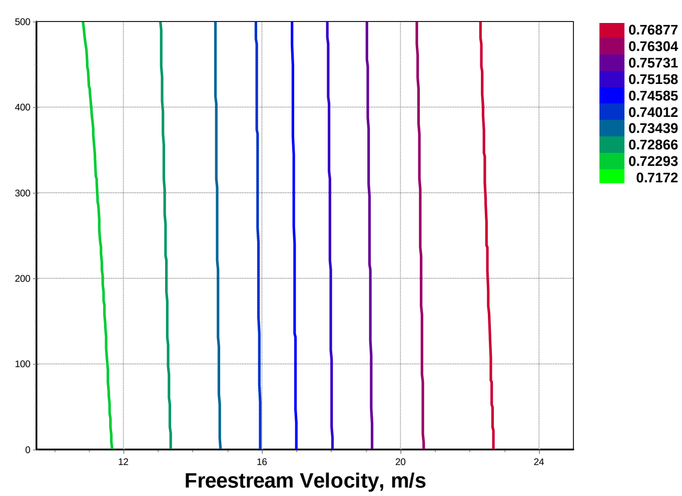

Figure 29. Motor efficiency vs. velocity and altitude.

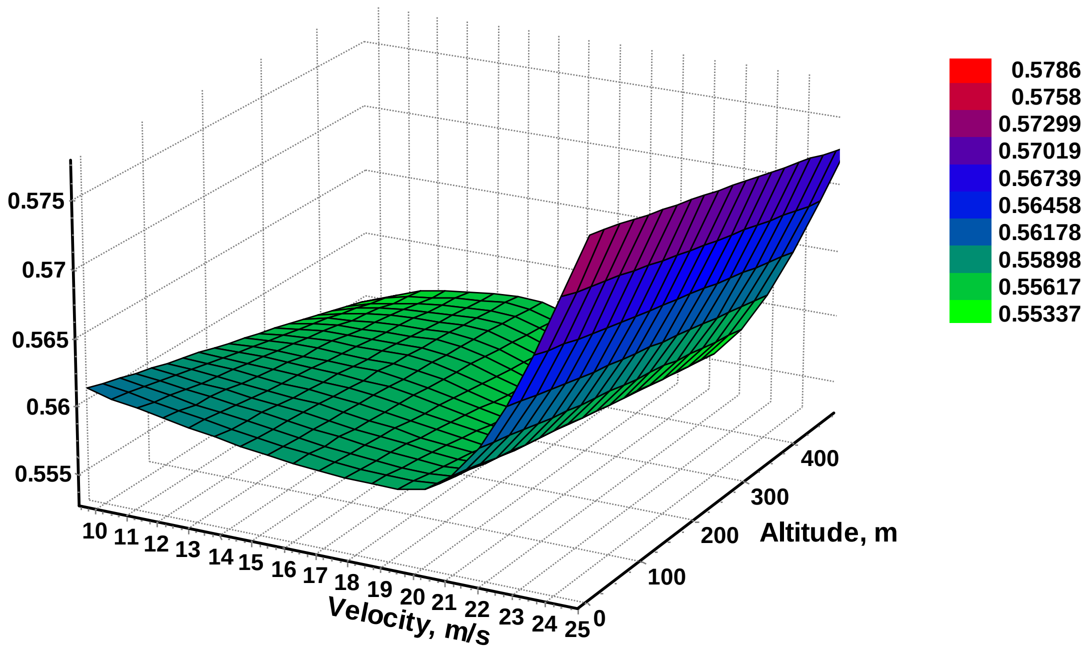

Figure 30. **Propeller efficiency vs. velocity and altitude, carpet plot.**

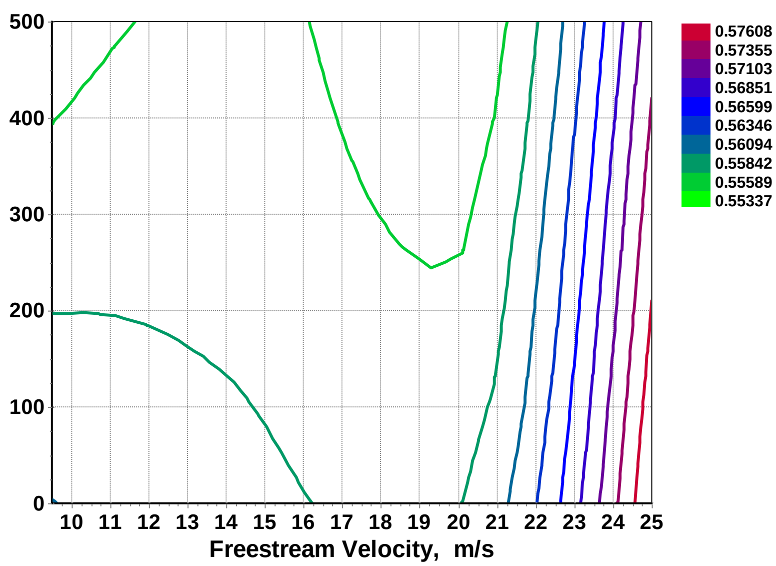

Figure 31. **Propeller efficiency vs. velocity and altitude.**

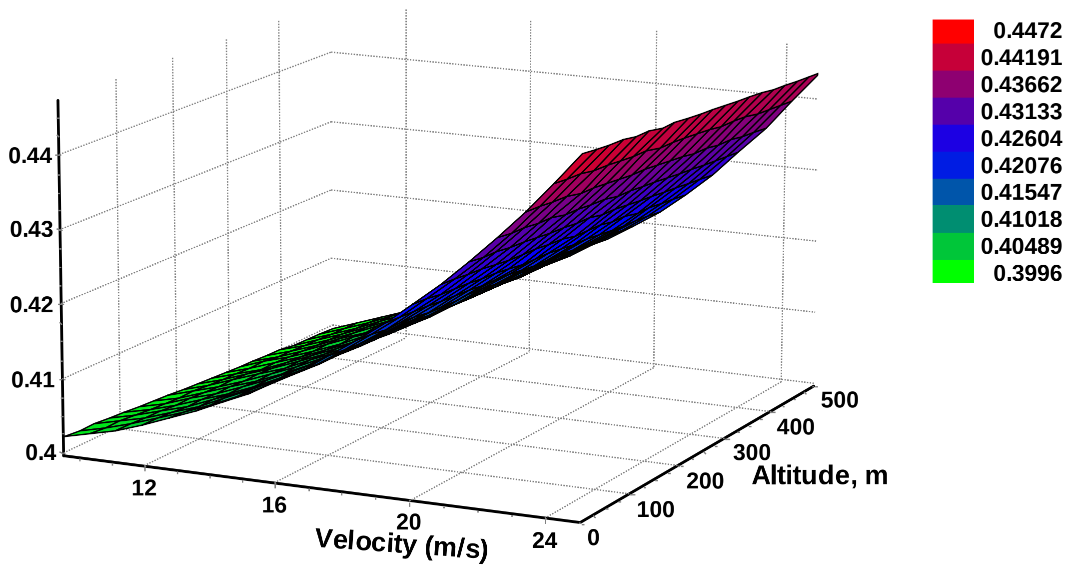

Figure 32. Overall efficiency vs. velocity and altitude, carpet plot.

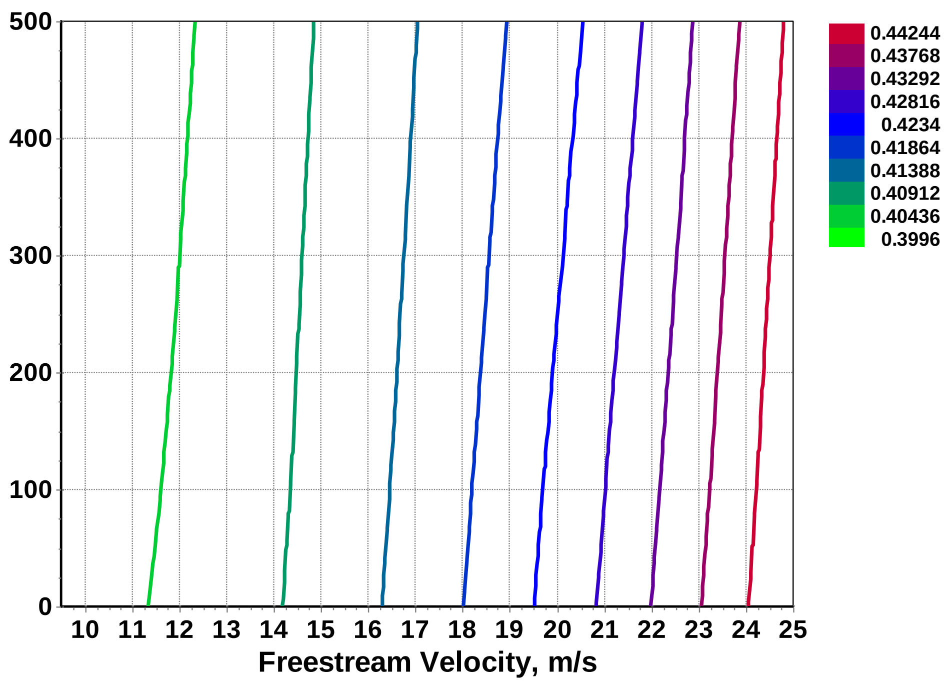

Figure 33. Overall efficiency vs. velocity and altitude.

In order to better understand the importance of span efficiency, it is plotted against velocity and altitude in Figure 34 and Figure 35. It is shown that peak efficiency occurs near the endurance velocity, and decreases as speed increases. With higher speed, the coefficient of lift decreases and induced drag increases, so Eq. (8) calls for a decreased span efficiency.

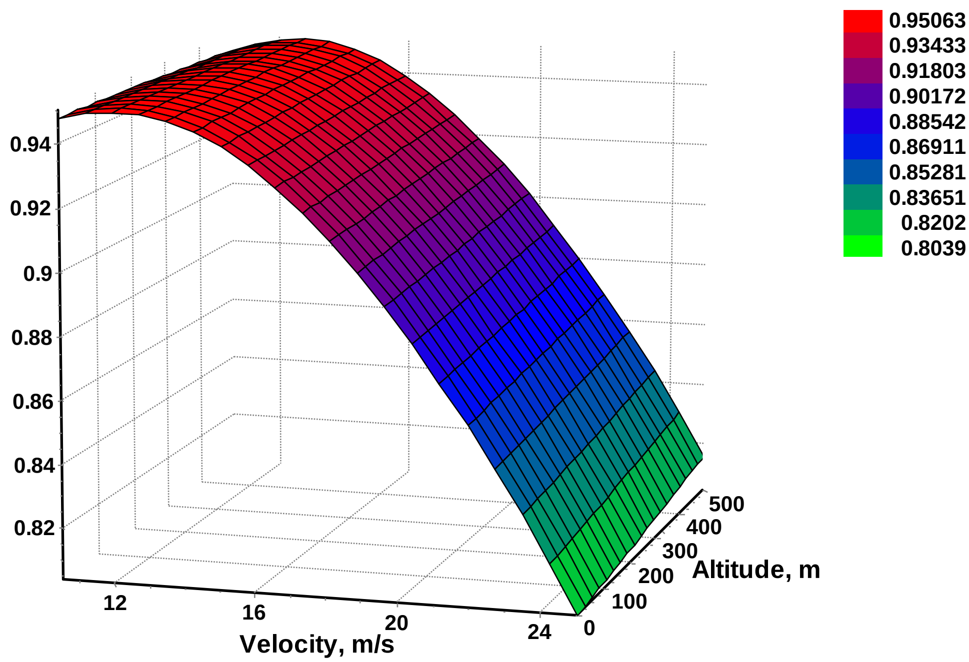

Figure 34. Span efficiency factor vs. velocity and altitude, carpet plot.

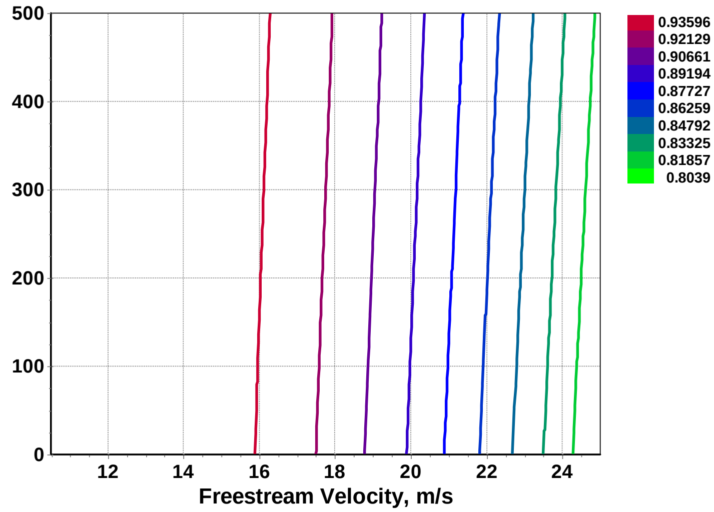

Figure 35. Span efficiency factor vs. velocity and altitude.

Airframe power-required follows the expected trend shown in Figure 36 and Figure 37. Minimum power-required is achieved at the endurance velocity and increases in a cubic nature. Increased altitude is demonstrated to have a slightly higher cost at high velocities, but plays a relatively minor role.

Overall total electrical power-required simply reflects the inefficiencies of the motor (Figure 28) and propeller (Figure 30) cast against airframe power-required (Figure 36). The plots of Figure 38 and Figure 39 show the same trend as the airframe power required, but with increased magnitude.

This trade study of velocity and altitude demonstrated that of these two variables, velocity is significantly more important than altitude. This is as expected because the power-required Eqs. (42) and (48), include velocity raised to the third power. The only effect of altitude is on density, which decreases with increasing altitude.

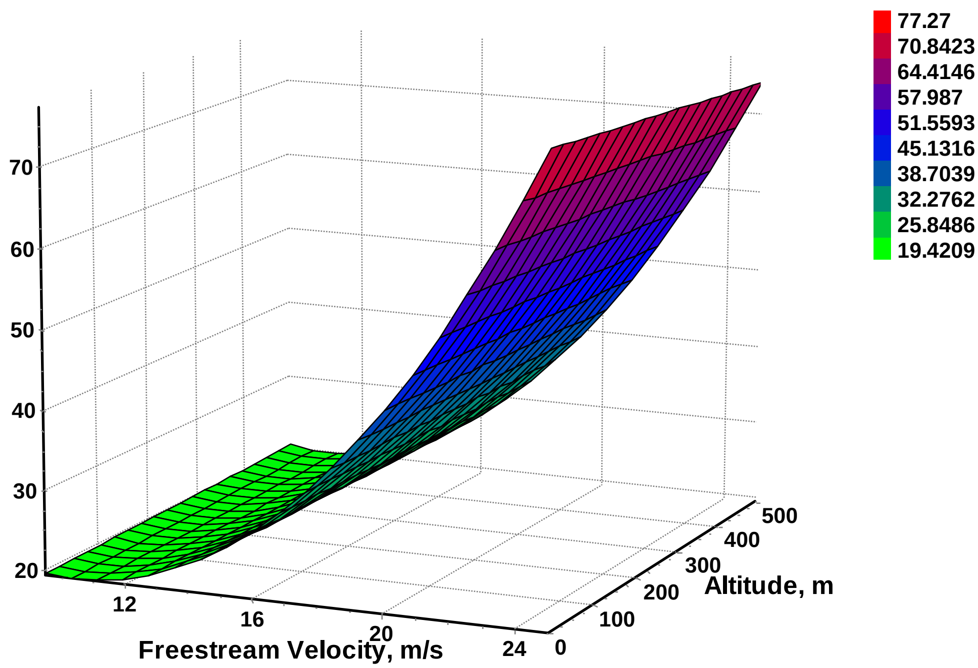

Figure 36. Airframe power required vs. velocity and altitude, carpet plot.

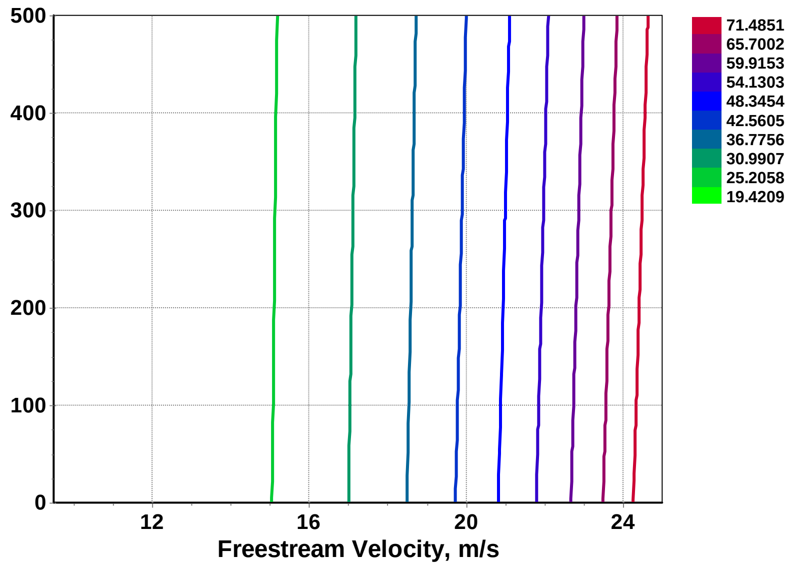

Figure 37. Airframe power required vs. velocity and altitude.

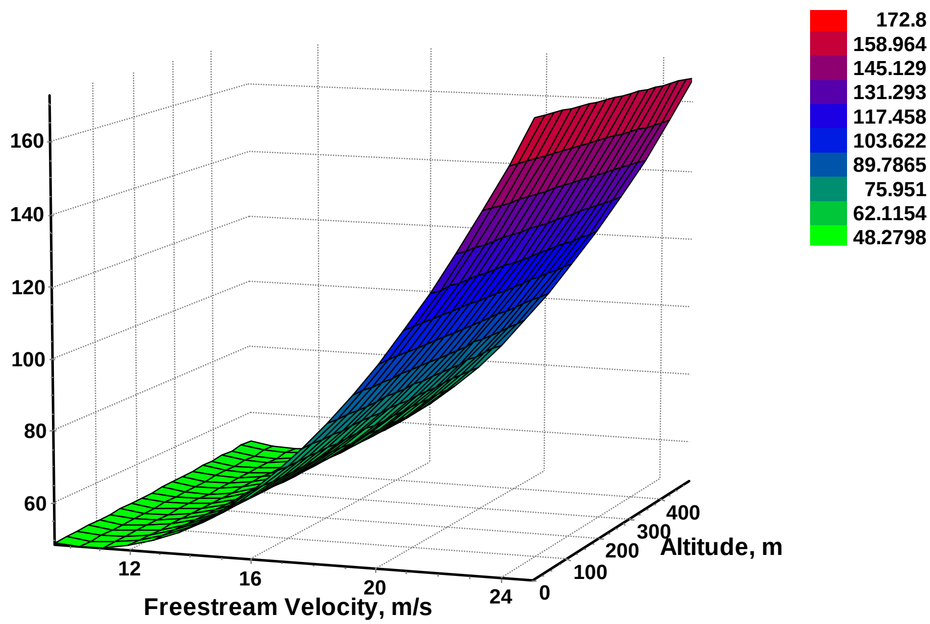

Figure 38. **Total electric power required vs. velocity and altitude, carpet plot.**

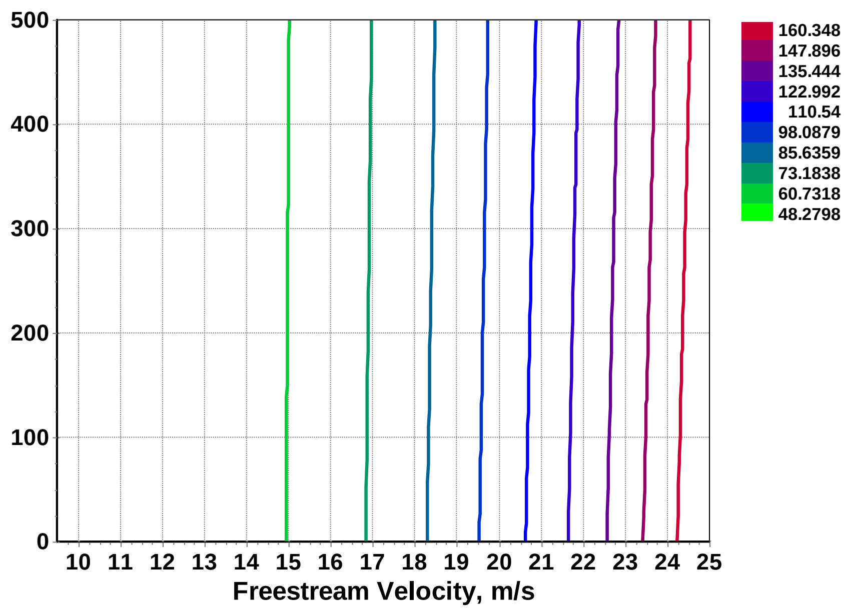

Figure 39. **Total electric power required vs. velocity and altitude.**

IV.2.2. Velocity and MAV Mass

The effect of altitude was analyzed because it is an important flight parameter. The effects of mass are now shown because mass is arguably the main design variable for a new flight vehicle. Figure 40 through Figure 49 demonstrate that when compared to velocity, mass plays a minor role in the observed outputs. Figure 40 and Figure 41 show motor efficiency increasing from 69% to 77% as velocity increases for the light configuration. The heavier configuration shows an increase from 73% to 77%. This increase is anticipated up to a certain design point, but efficiency should then decrease to zero as the motor reaches stall torque at maximum current.

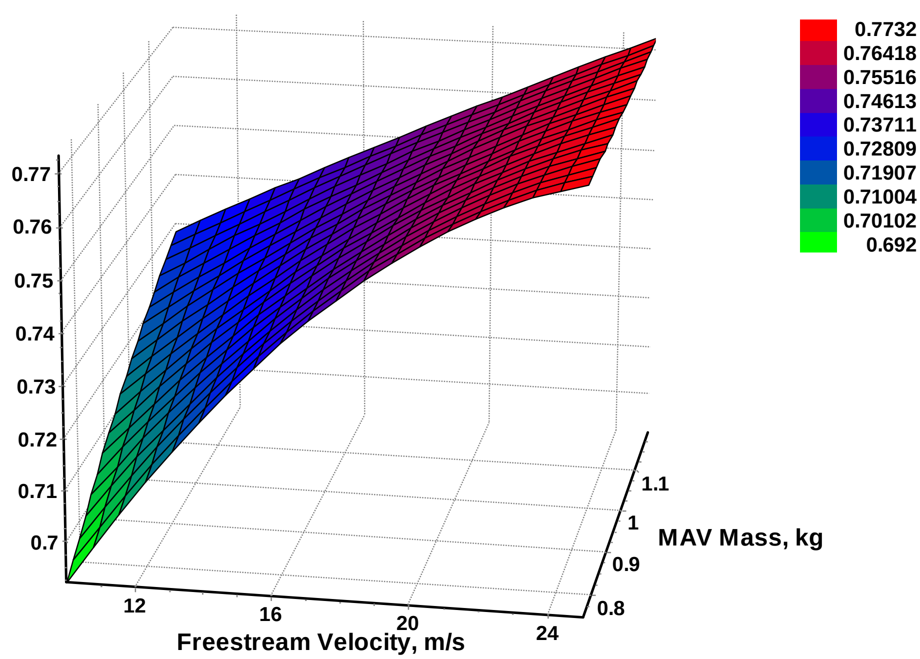

Figure 40. Motor efficiency vs. velocity and mass, carpet plot.

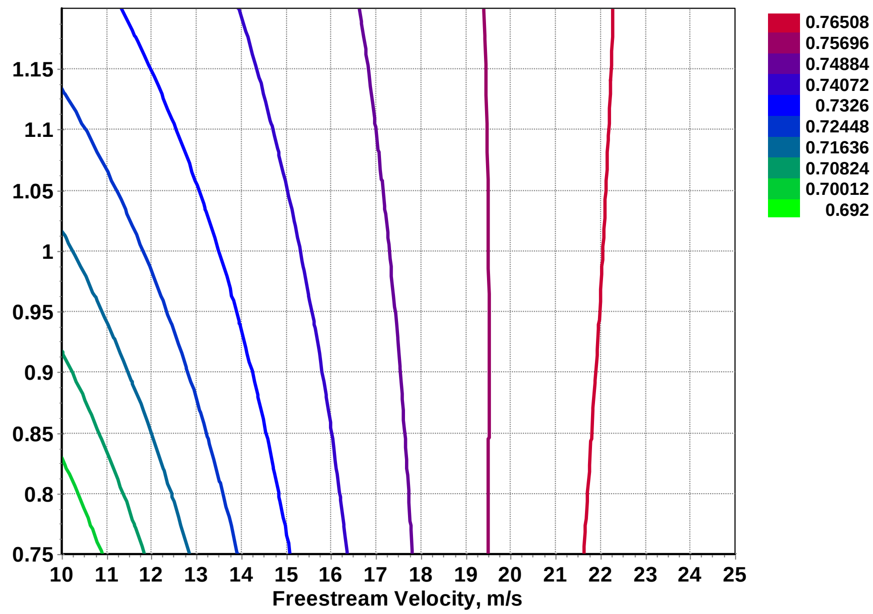

Figure 41. Motor efficiency vs. velocity and mass.

Even though the GenMAV propeller and motor selection were the result of collective design experience, the components were not rigorously matched to the airframe or each other. Furthermore, the results shown here illustrate an ill-conceived propeller geometry model that clearly does not reflect an optimal selection. The results of Figure 40 and Figure 41 illustrate an insufficient torque requirement on the motor. As less mass would drive a lower torque requirement at slower speeds, a motor efficiency minimum of 69% is conceivable. The propeller results shown in Figure 42 and Figure 43 reflect the same trend as the propeller efficiency verses velocity plot of Figure 30, where a local maximum efficiency is found near the endurance speed, with another high efficiency region at higher speeds. However, the advance ratio plot of Figure 27 suggests that the higher velocities may not reflect a realistic condition. Another reason why this may be

true is that propellers generally perform worse at higher speeds, but the propeller illustrated here has best performance at the highest velocity.

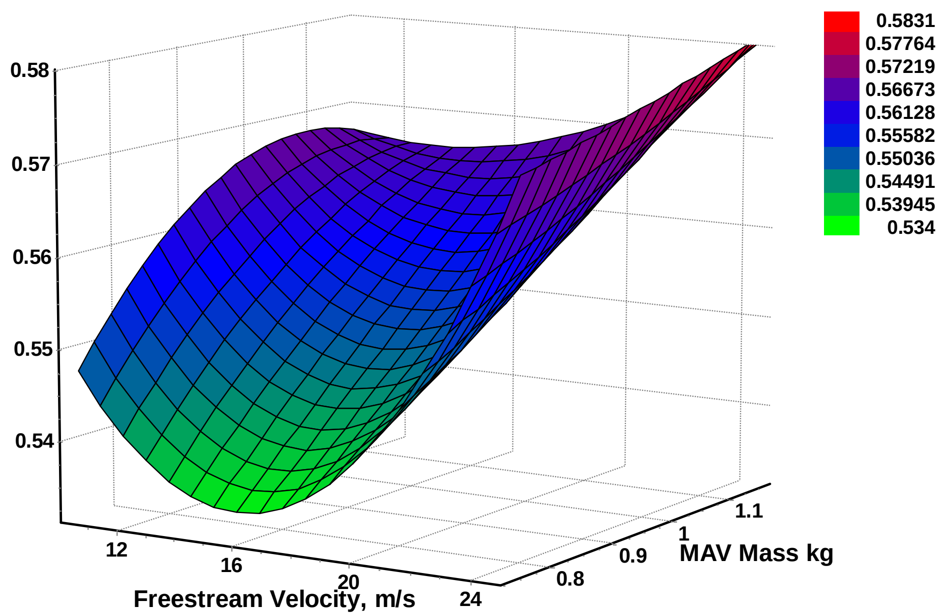

Figure 42. Propeller efficiency vs. velocity and mass, carpet plot.

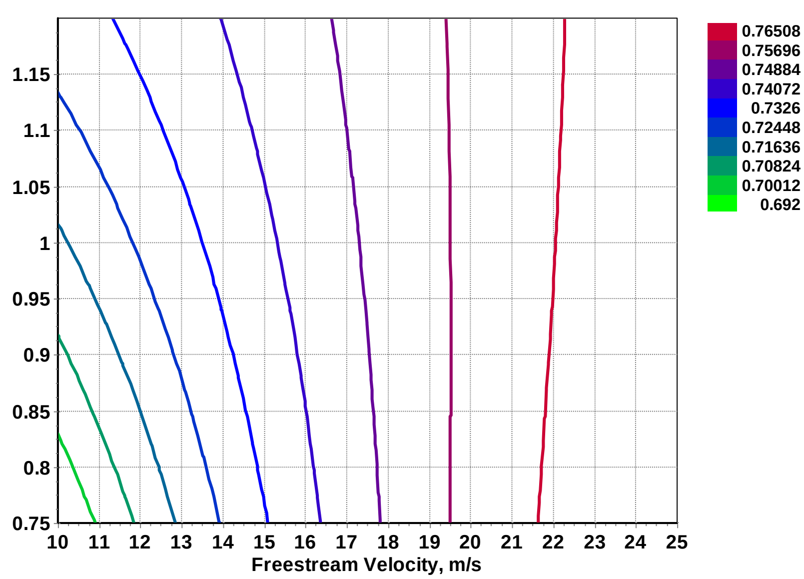

Figure 43. Propeller efficiency vs. velocity and mass.

Propeller efficiency is shown here to be more dependent on mass at lower speeds than at higher speeds, where the plots show a maximum of approximately 58%. Figure 44 and Figure 45 reflect the fact that overall efficiency is the product of the motor and propeller efficiencies. These plots demonstrate that mass plays a much greater role at low velocities than at higher velocities.

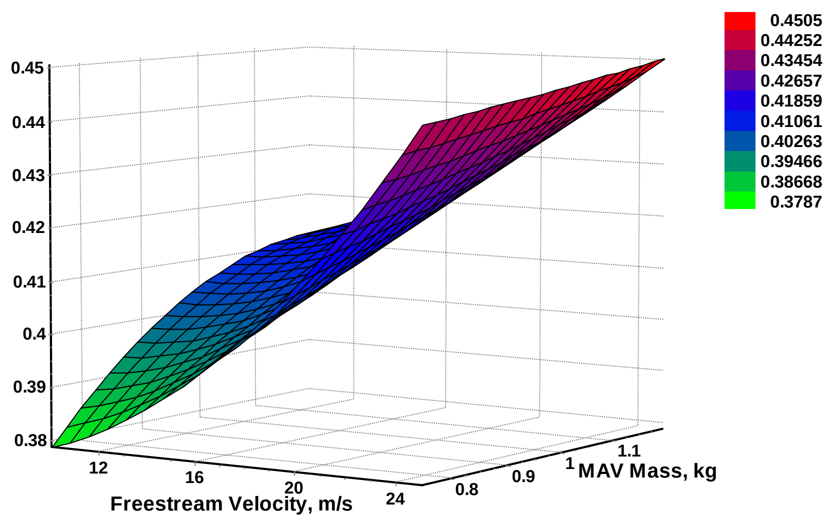

Figure 44. Overall efficiency vs. velocity and mass, carpet plot.

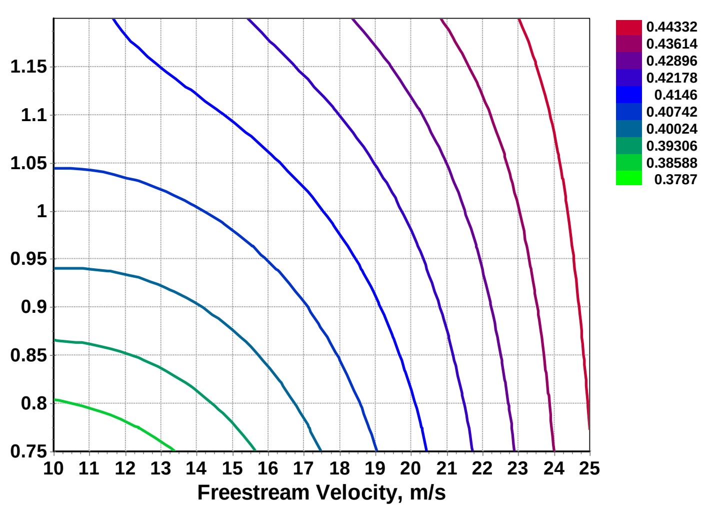

Figure 45. Overall efficiency vs. velocity and mass.

Airframe power-required and total power required exhibit the same behavior in Figure 46 through Figure 49 as previously shown in the velocity-altitude trade study. In the case of mass, the required power is noticeably different between low mass and high mass at slow velocities. At higher velocities, since a tremendous amount of relative power is required to overcome drag, the mass is less of a concern.

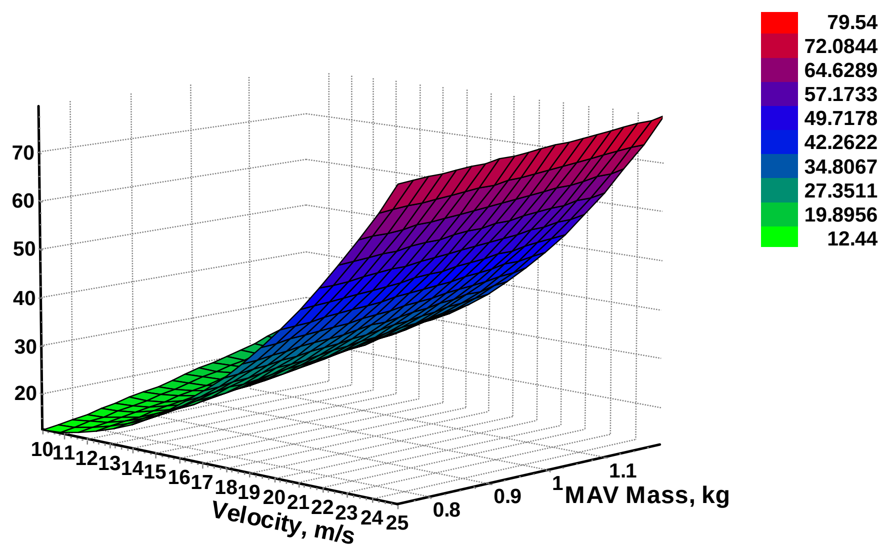

Figure 46. Airframe power required vs. velocity and mass, carpet plot.

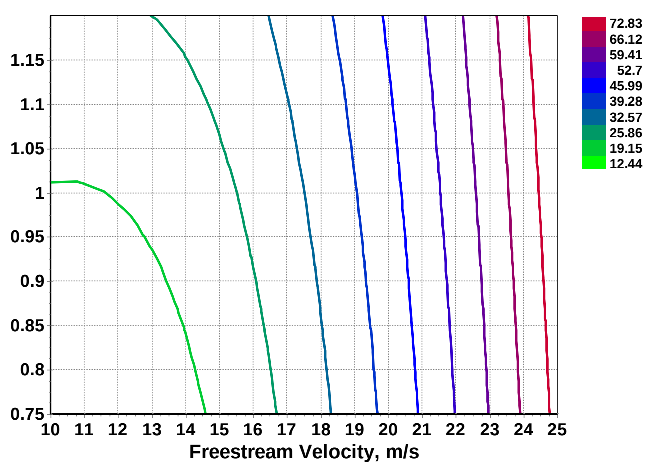

Figure 47. Airframe power required vs. velocity and mass.

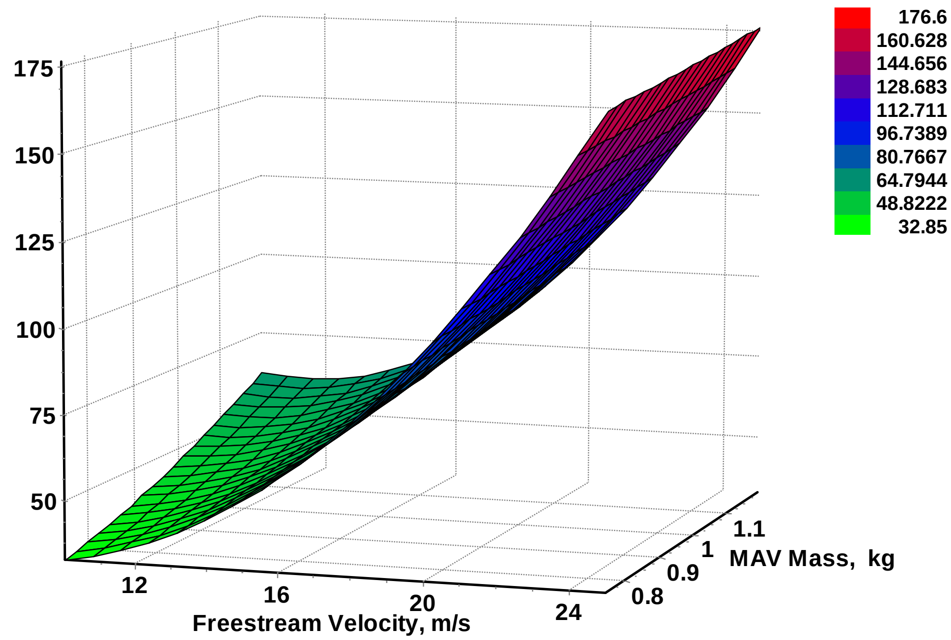

Figure 48. **Total electrical power required vs. velocity and mass, carpet plot.**

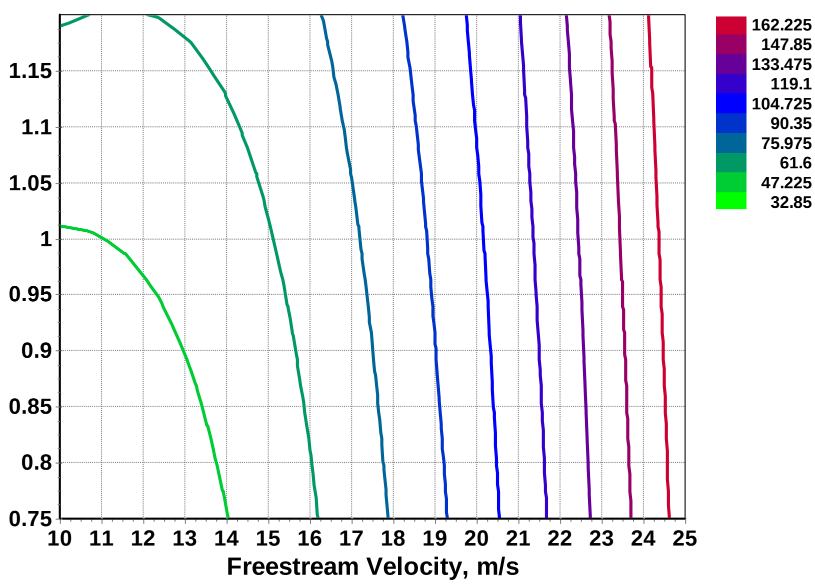

Figure 49. **Total electrical power required vs. velocity and mass.**

IV.2.3. Velocity and Rate of Climb

Nearly the same results illustrated for the velocity-mass trade study are reflected in the velocity-rate of climb study shown by Figure 50 through Figure 59. This is because where mass drives an aircraft wing loading requirement, rate of climb drives a wing loading performance requirement that results from a higher required lift.

Again, it is seen in Figure 50 and Figure 51 that the motor efficiency varies greatly at low speed depending on the rate of climb, but is rather constant at increased velocity independent of the rate of climb. However, the slope at peak velocity appears to have decreased compared to Figure 28 and Figure 40 of the previous studies, which may suggest that a maximum motor efficiency of 77% is nearly reached.

Propeller efficiency shown in Figure 52 and Figure 53 suffers at a descending rate of climb because the propeller is not loaded. At higher rates of climb and higher velocities, the propeller nears the expected 60%. At a rate of climb of zero, the same value of approximately 57% is displayed. This matches Figure 27.

Overall efficiency, shown in Figure 54 and Figure 55 demonstrate the combined results from the motor and propeller efficiencies. A rate of climb increase from level flight to 1 m/s requires approximately 50% more power, as shown in the airframe total electric power-required plots of Figure 56 through Figure 59.

The total electrical power-required of Figure 58 and Figure 59 displays a minimum near the endurance velocity, which is typical for an electric motor/propeller aircraft.

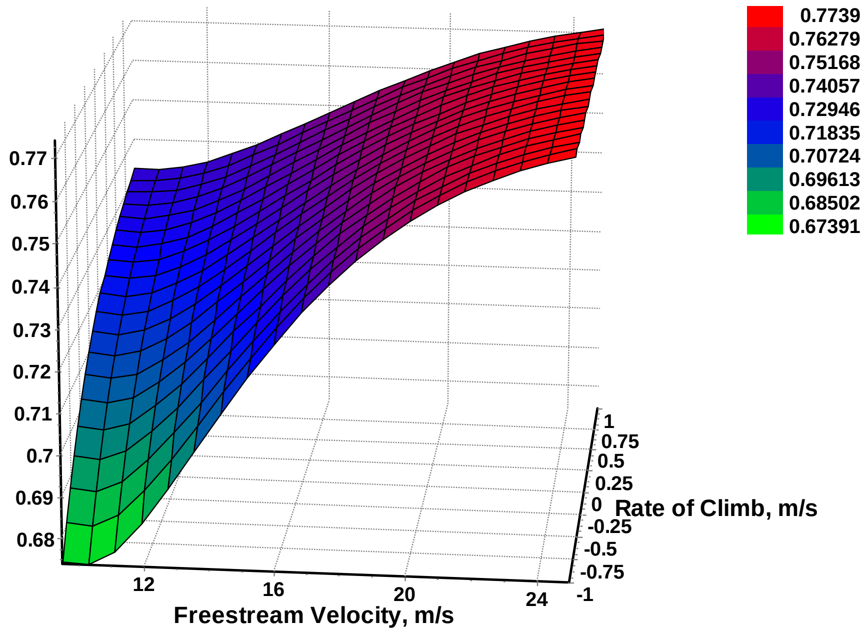

Figure 50. Motor efficiency vs. velocity and rate of climb, carpet plot.

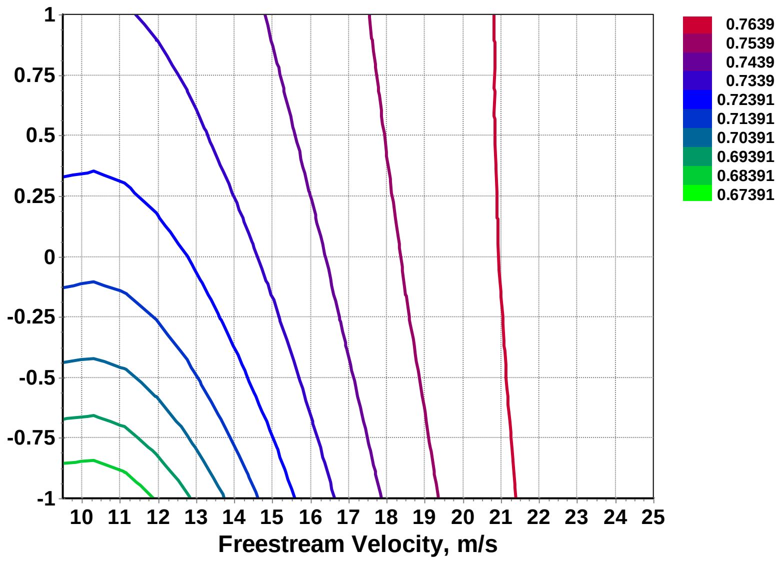

Figure 51. Motor efficiency vs. velocity and rate of climb.

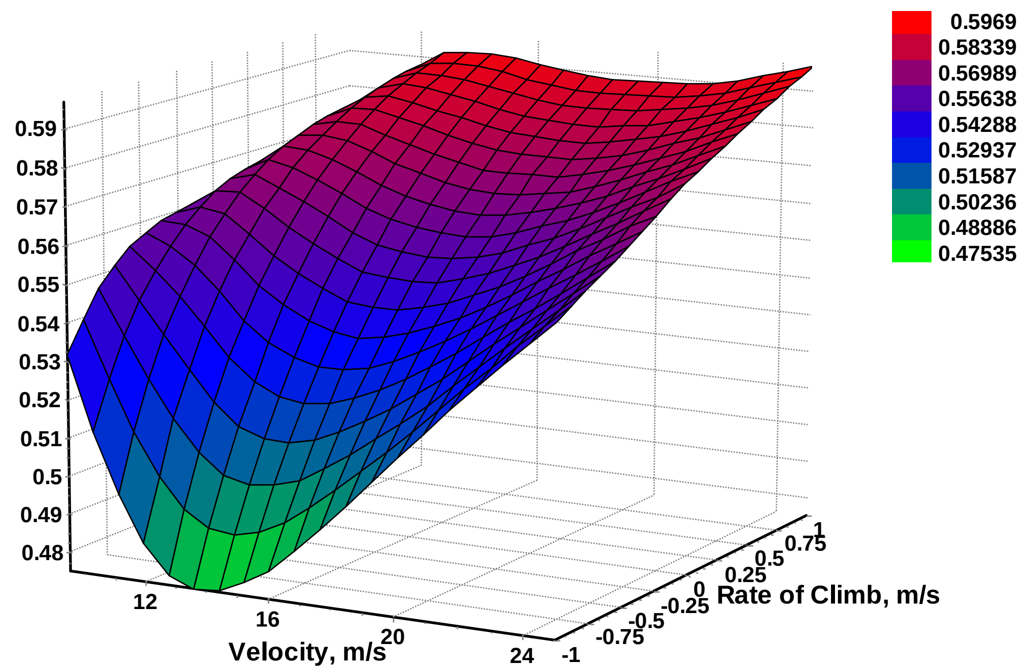

Figure 52. Propeller efficiency vs. velocity and rate of climb, carpet plot.

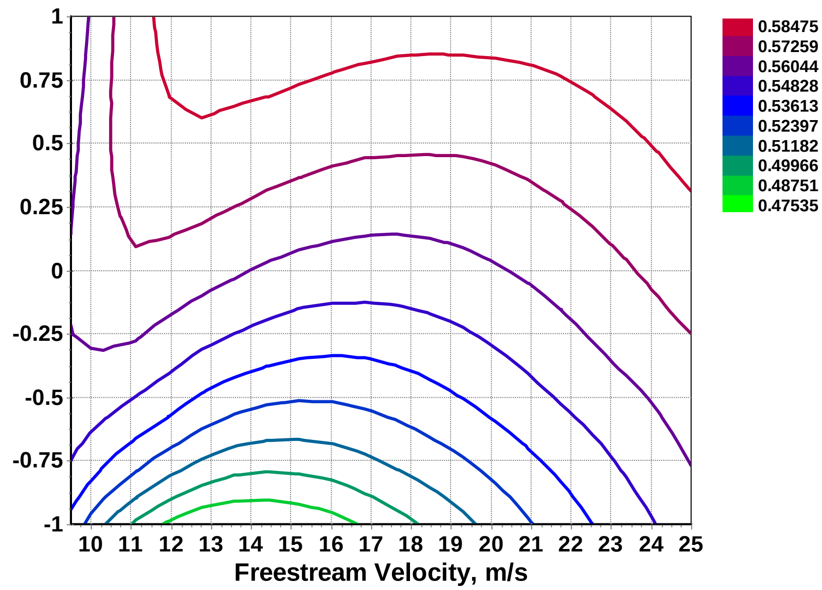

Figure 53. Propeller efficiency vs. velocity and rate of climb.

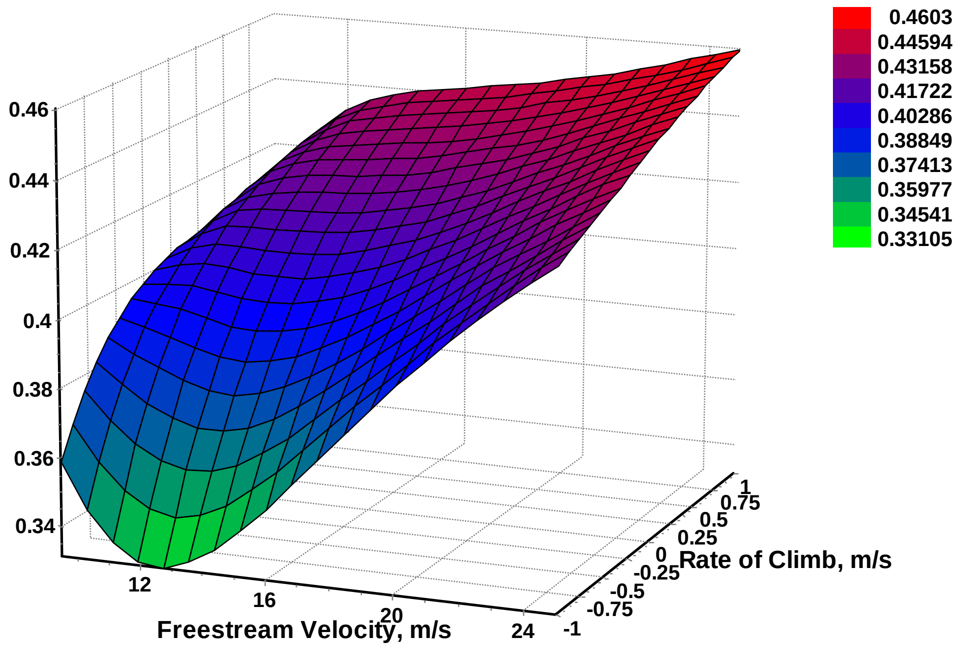

Figure 54. **Overall efficiency vs. velocity and rate of climb, carpet plot.**

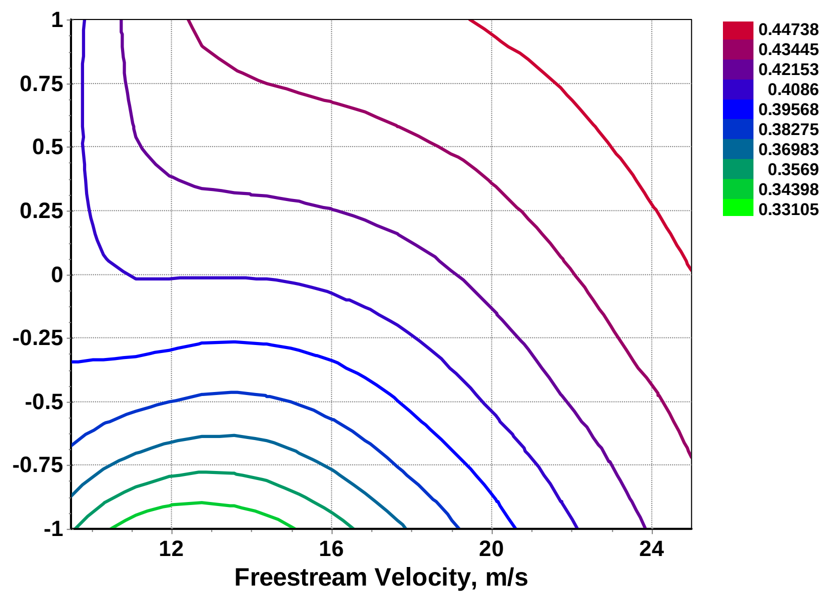

Figure 55. **Overall efficiency vs. velocity and rate of climb.**

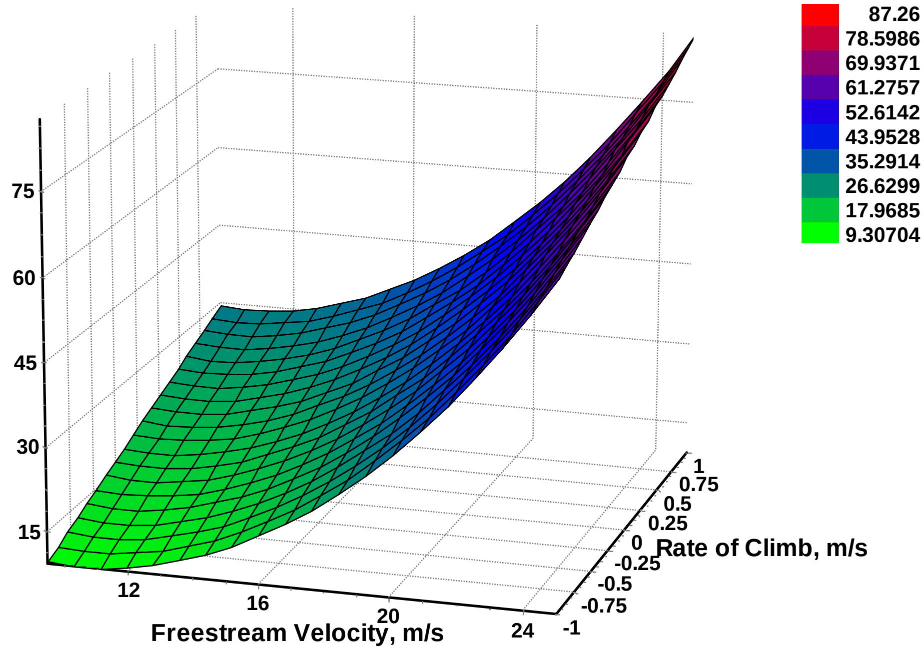

Figure 56. **Airframe power required vs. velocity and rate of climb, carpet plot.**

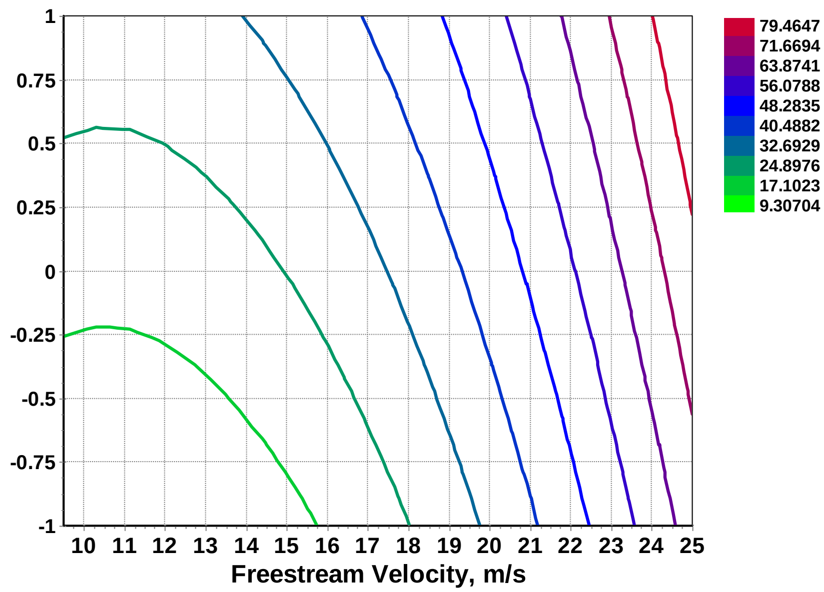

Figure 57. **Airframe power required vs. velocity and rate of climb.**

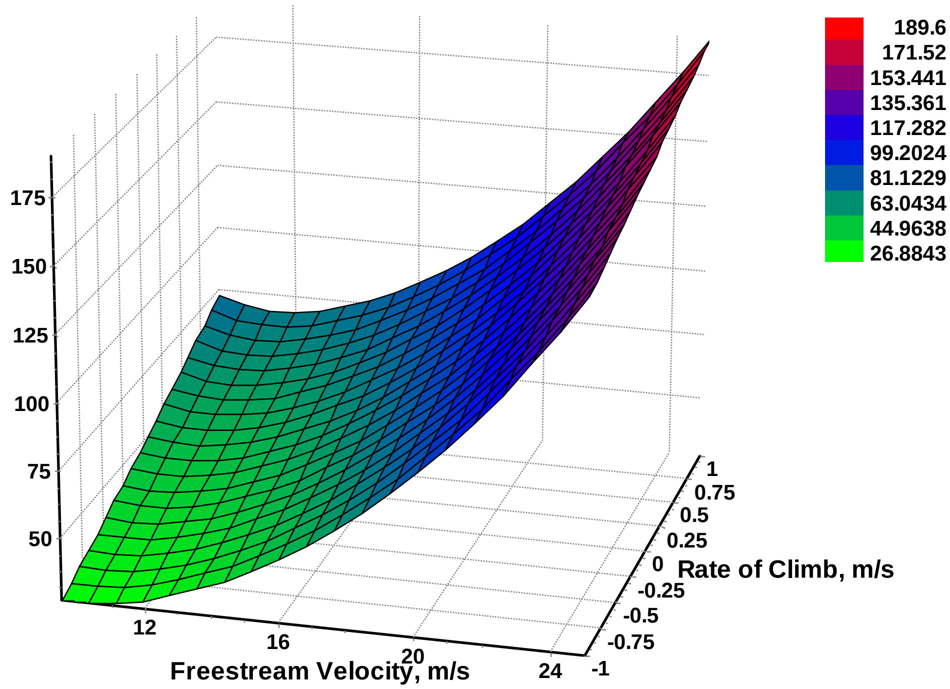

Figure 58. Total electric power req'd vs. velocity and rate of climb, carpet plot.

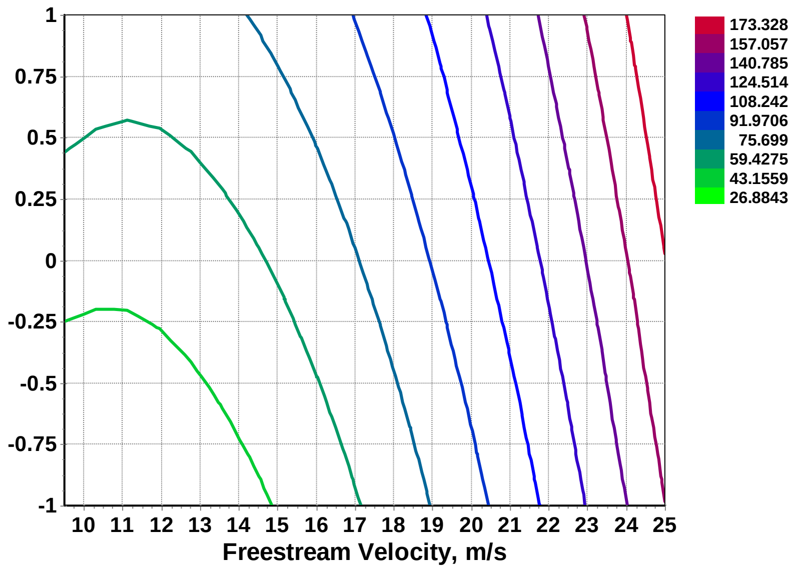

Figure 59. Total electric power required vs. velocity and rate of climb.

This section highlighted the means in which trade studies can identify or confirm relationships between design variables. Parametric carpet plots confirmed that the airframe aerodynamic model and the propulsion model provide expected results over a range of inputs. The next section will bring together these components to simulate a mission.

IV.3. Mission Simulations

Even though various parts of the MAV can be optimized for specific flight conditions, the real intent of the work is to optimize the MAV power source selection for a set mission. This mission includes defined durations, altitude designations, and defined speeds for each mission segment. Several mission profiles that look like the baseline mission profile of Figure 11 were analyzed.

The first simulation was for a typical mission requiring a climb to 500 m at a rate of climb of 1.5 m/s. It then cruised to station at a user specified speed of 17 m/s, loitered at endurance velocity, and returned at cruise velocity. Finally, it descended at 1 m/s. Because both the endurance and cuise velocities (10.22 m/s and 13.46 m/s) were below stall (13.97 m/s), both were increased by a small fraction to be above stall. From the power profile of Figure 60, it is seen that mission segments 3 and 4 reflect this situation. With the EDS taking 50% of the average power load, the battery uses less and less as the mission elapses.

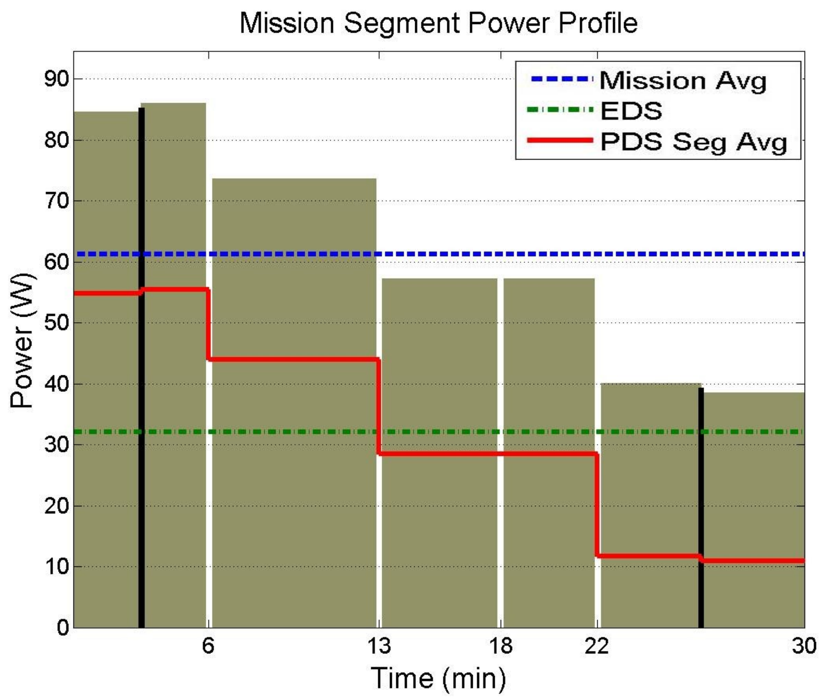

Figure 60. Power profile for P_{EDS} =50% P_{avg} , C_{Lmax} = 1.16.

In order to inspect the program's ability to actually employ the max range speed, the maximum lift coefficient was raised from 1.16 to 1.30. This had the desired effect of reducing the stall speed to 13.19 m/s, which allowed the MAV to fly at its true cruise velocity, but not at loiter velocity as shown by segments three and four of Figure 61.

When the fuel cell percentage is reduced from 50% to 32%, the battery must carry more of the load as shown in **Error! Reference source not found.**. Each of these plots exhibit the expected behavior and the power requirements are within reason. The user would only have to input desired flight envelope and power parameters, and the entire simulation will run automatically.

Figure 61. Power profile for P_{EDS} =50% P_{avg} , C_{Lmax} = 1.30.

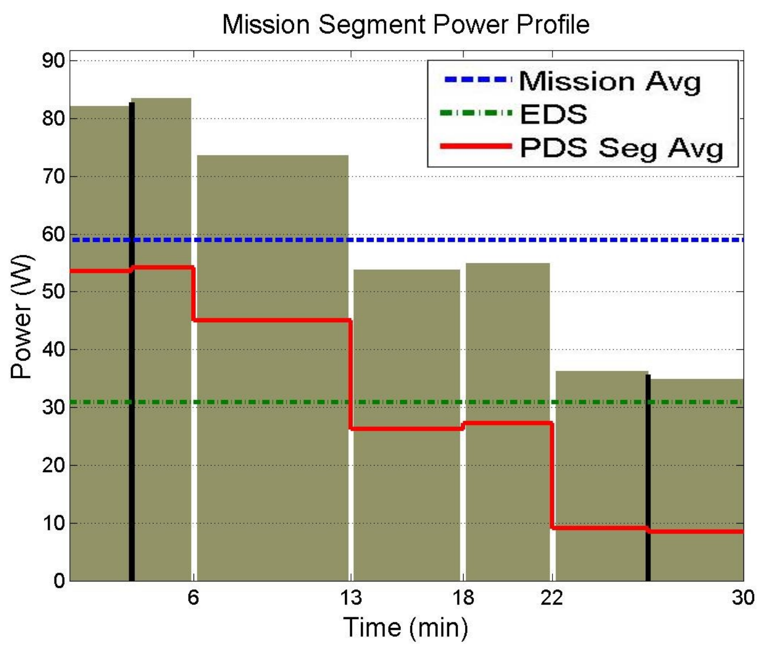

Figure 62. Power profile for P_{EDS} = 32% P_{avg}, C_{Lmax} = 1.30.

With the models established, the next set of analysis looks at varying the power system components in order to reduce mass while still satisfying mission requirements.

IV.4. Power System Design Optimization

Perhaps the most insightful method of running this conceptual design tool is by performing optimizations. Since this effort was built for the power designer, a useful analysis is determining how to size the fuel cell with respect to the battery. Energy requirements are highly dependent upon mission and duration, so this investigation used the same mission from Figure 60 with the original maximum lift coefficient of 1.16. In all cases, the PDS was not required to maintain a certain SOC, and there were no recharging requirements. This section continues to use the notation defined by Table 5.

IV.4.1. Analysis Overview

The analysis is divided into four distinct sections. The first step was to sample the design space by running a full-factorial Design of Experiments (DOE) study. This technique tests specific points to see if they are valid or invalid design points. The term *valid* is used here to mean that a design point meets a certain criteria, such as total power system mass under a certain value. Results of the samples often suggest where optimal design points may exist. The DOE also identifies the relative importance of each of the design variables in the study as compared to the objective. Two families of solutions were identified by the DOE. The second part of the analysis elaborated on one of the solution families identified by the DOE, and the peak energy capability of the PDS was limited to 160 and 200 W-hr/kg. A maximum PDS_{SE} of 160 W-hr/kg correlates to a current technology Li-Po battery, and a maximum PDS_{SE} of 200 W-hr/kg correlates to

near-term Li-Po battery. The five design variables shown in Table 8 were optimized and the minimum power system mass hybrid configuration was identified. The third analysis set PDS and EDS values based on practical current and near-term properties of batteries and fuel cells. Here, differential was the only variable optimized to determine hybridization potential. The four families of results are each explained. The forth set of analysis explores how power source component values and their best relative sizing change when the mission is extended from 30 to 60 minutes.

The Model Center Gradient Optimizer allows for single or multiple design variable optimizations to meet linear or nonlinear cost functions and constraints. Instead of creating a rigorous mathematical cost function, the user poses the optimization problem through a simple GUI where variables are dragged and dropped. Bounds and starting guesses can be set. Advanced options include a limit on function evaluations and iterations. Constraint, objective function, and projected gradient tolerances have a default condition, or can be set by the user. Model Center decides how to solve the optimization, but informative text file outputs allow the user to see many of the evaluations and choices made at each iteration. Some of the methods used are the quadratic, cubic, and bisection line search. Tolerances proved to be very important. After some investigation, the default tolerances were found to be too large, which resulted in convergence to less desirable local minimums. When this happened, Model Center still returned feasible points, but they were not optimal. Although more time was required, better results were obtained by decreasing the default tolerances by on order of magnitude to those shown in Table 7.

Table 7. Optimization tolerances used for analysis.

Finite difference step size for calculating gradients	0.0001
Tolerance on objective function	0.0001
Tolerance on constraint feasibility	0.0001
Tolerance on projected gradient	0.0001

IV.4.2. Analysis 1: Design of Experiments

First, the design space was evaluated by executing a three-level full-factorial Design of Experiments (DOE) with the constraints shown in Table 8. The values chosen represent either current or near-term battery and fuel cell possibilities. A DOE runs all design variable permutations and identifies which variables most affect the design. In this DOE, the three levels are the minimum value, middle value, and maximum value of each variable, which results in 243 runs. For example, differential was only tested at 0%, 67%, and 133%. This study illustrates the tradeoff between resolution, and ability to sample the entire design space. Results are shown by glyph plots in Figure 63 and Figure 64. A table of DOE results is provided in Appendix C.

Table 8. Input constraints for three-level full factorial DOE study.

	Units	Minimum	Maximum	Representation
EDS_{SP}	W/kg	10	500	Small to Large FC
EDS_{SE}	W-hr/kg	500	1000	Small to Large FC
PDS_{SP}	W/kg	1200	6000	Current Li-Ion
PDS_{SE}	W-hr/kg	60	200	Near-Term Li-Po
Differential	Percent	0	133	No Hybrid to Full Hybrid

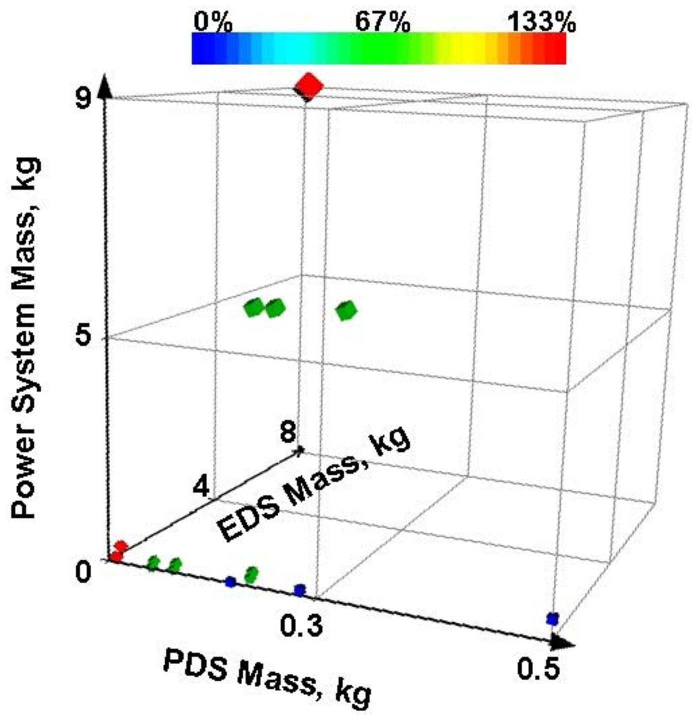

Figure 63. Glyph plot of DOE results.

The glyph plot representing a sample of the design space is shown in Figure 63. This is a five dimensional plot. The horizontal axes represent the component masses of the individual EDS and the PDS. Total power system mass is shown on the vertical axis. Each point represents a potential design. The forth dimension is shown by the magnitude of each point, consistent with its total power system mass. The level of hybridization is represented by color, which is the fifth dimension. Blue is single component PDS system. Red is a two-component system with the EDS sized to be 133% of the average mission power. Often, a variety of inputs will yield the same result, so most points shown on the plots represent families of feasible points. This provided many invalid

designs, but when a more narrow view near the origin is taken, the minimum mass design points are apparent as shown in Figure 64.

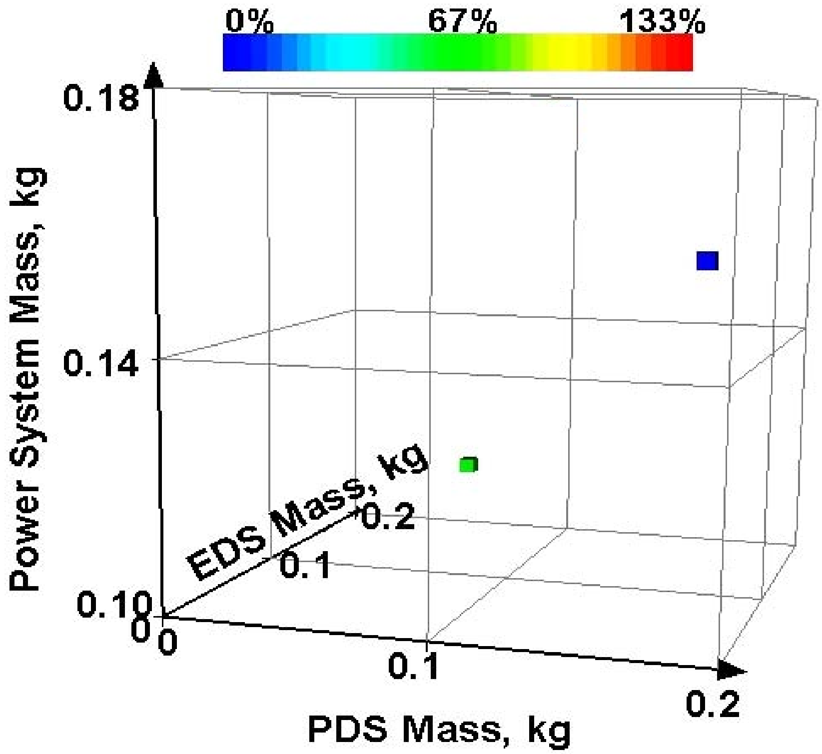

Figure 64. Glyph plot of reduced design space.

The minimum mass design points are actually families of approximately 10 points that yield the same value. All total power mass values include a 10% packaging factor and all analysis is done on the same half-hour mission and the same platform. The mission required power was determined based on the actual power system mass of 145 g, and does not recalculate mission power resulting from a changed power system weight.

- *DOE Solution Family One*: These solutions show that a hybrid system can result in a reduced power system mass, and are represented by the lowest green point in Figure 64. Use the EDS at a power value of 67% of average mission power. If a 200 W-hr/kg PDS_{SE} is available, design will be driven to and constrained by this.

The minimum PDS_{SP} of 1200 W/kg is more than enough. Power system mass was the minimum found of all DOE test points: 147 g. However, this design chose to maximize EDS_{SP} at 500 W/kg. Note that this split may not be optimal because the design point of 67% was technically a "user-input," as provided by the DOE. Furthermore, maximum power and energy cannot be attained simultaneously, so values used within the DOE might not reflect real world options. Optimality will be explored in IV.4.3, but the power system mass is reduced as compared to the PDS battery only.

- *DOE Solution Family Two*: The second lowest point of Figure 64 is blue, which has a differential of 0%, and cannot use the EDS to reduce power system mass. The design will be driven to a 200 W-hr/kg PDS_{SE} if available. The energy of the PDS will still be the bottleneck. The minimum PDS_{SP} of 1200 W/kg is more than enough. Plenty of power capability exists, so mass is determined by energy. Total power system mass was 168 g.

The DOE identified EDS_{SP} as the primary variable that affects power system mass minimization. These results are shown in Figure 65. The split between the EDS and the PDS, which is the differential, proved to be the next most important. The value chosen for PDS_{SE} only slightly affects the design, while the ranges of EDS_{SE} and PDS_{SP} are high enough in most cases that their value does not affect the design at all. The above DOE study provided general information on the design space, but did not focus on the most feasible designs. The next step was to compare what a maximum PDS_{SE} of 160 W-hr/kg could do verses 200 W-hr/kg.

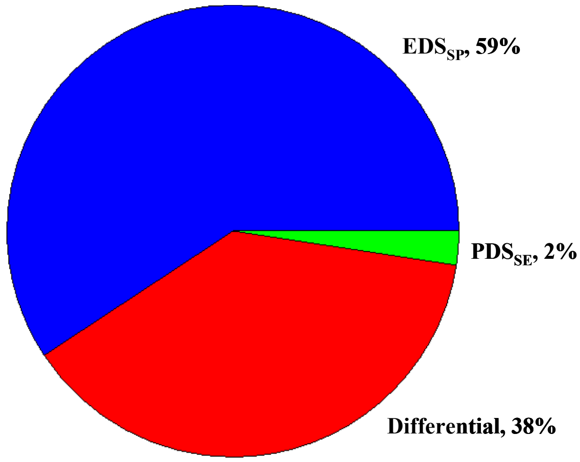

Figure 65. Relative importance of each variable to mass minimization

IV.4.3. Analysis 2: Compare Current and Near-Term Li-Po Possibilities

This part of the analysis accomplished two things. First, it expanded on Solution Family One by allowing a five-design variable optimization routine to determine the best hybridization split. Second, it looked specifically at a maximum PDS_{SE} that reflects current Li-Po capability. The previous DOE values for PDS_{SE} were 200 W-hr/kg, where 160 W-hr/kg would be more realistic. Except for the PDS_{SE} changes here and the limitation on PDS_{SP} to 1200 W/kg, the other design variables (EDS_{SP}, EDS_{SE}, and differential) had the limits shown in Table 8. Results confirmed PDS/EDS hybridization has the potential to reduce power system mass as shown in Table 9. A higher specific energy PDS would yield a lower power system mass for a hybrid system. For reference, the Li-Po only system mass with packaging factor resulted in 211 g. The hybridized power system with the near-term battery offers a reduced power system mass as compared with that of the current Li-Po battery.

Table 9. **Hybridization results using different PDS_{SE} values.**

	Units	Current Li-Po	Near-Term Li-Po
Maximum PDS_{SE}	W-hr/kg	160	200
Mission power max	W	86	86
Mission power average	W	61	61
Mission energy	W-hr	31	31
Differential	%	90	65
PDS mass	g	143	133
Total power system mass	g	157	147

The optimization routine was set to vary the differential between 0 and 133% in order to minimize power system mass. The optimized differential for minimum power system mass for the current Li-Po hybrid is shown in Figure 66. For this set of input parameters, the PDS is sized for depletion by the end of the mission when the EDS then must carry the entire load. The differential split between each scenario reflects the importance of component properties.

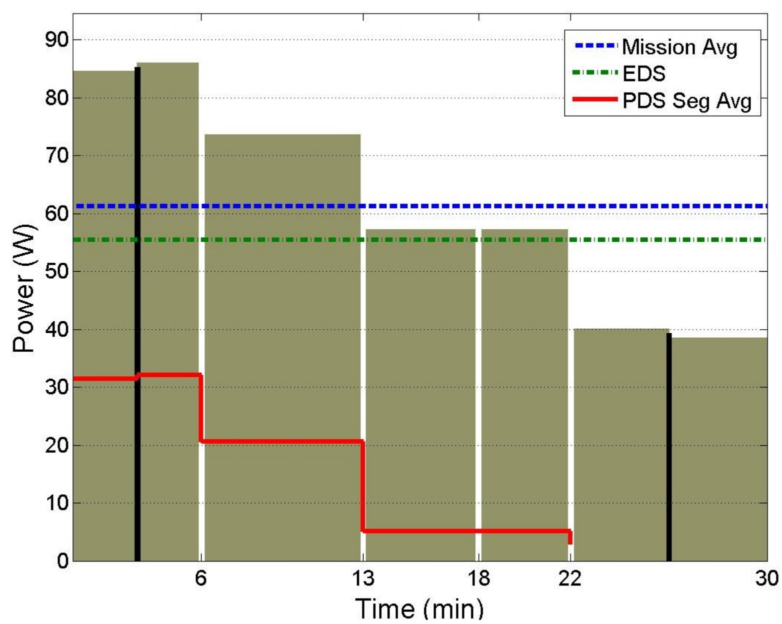

Figure 66. Optimized PDS/EDS split mission power profile.

IV.4.4. Analysis 3: Sample of Practical Power Source Property Inputs

Next, twelve cases of practical importance were analyzed. These represent best current technology Li-Po and Li-Ion verses properties loosely based on a full-scale ground-based fuel cell, properties of a fuel cell sized for a large UAV, and properties for a fuel cell sized for a MAV. Li-Po values of 1200 W/kg and 160 W-hr/kg are used for the PDS of runs 1-2 and 5-8. Li-Ion values of 6000 W/kg and 60 W-hr/kg are used for the PDS of runs 2-4 and 9-12. A futuristic fuel cell with properties of similar order of magnitude as a ground-based fuel cell represent the EDS of runs 1-4. When considering how this fuel cell would need to be scaled down for MAV application, it is the specific power value used here that make it futuristic. Fuel cell values of 100 W/kg and 500-1000 W-hr/kg that are possible for use today in large UAVs represent the EDS of runs 5-6 and 9-10. Lastly, small or micro fuel cells with values of 10 W/kg and ambitious 500-1000 W-hr/kg are used for the EDS of runs 7-8 and 11-12.

EDS and PDS values were user-inputs, and Model Center fluctuated the differential sizing between 0% and 150% of the average mission power. Optimality was judged on minimum power system mass. Inputs and resulting power system mass are shown in Table 10.

Table 10. **Various cases run to determine best split between PDS and EDS.**

Run #	Result Family	EDS_{SP} [W/kg]	EDS_{SE} [W-hr/kg]	PDS_{SP} [W/kg]	PDS_{SE} [W-hr/kg]	Differential	Power System Mass [kg]
1	One	500	500	1200	160	93%	0.157
2	One	500	1000	1200	160	93%	0.157
3	Two	500	500	6000	60	122%	0.182
4	Two	500	1000	6000	60	122%	0.182
5	Three	100	1000	1200	160	0%	0.211
6	Three	100	500	1200	160	0%	0.211
7	Three	10	1000	1200	160	0%	0.211
8	Three	10	500	1200	160	0%	0.211
9	Four	100	1000	6000	60	0%	0.562
10	Four	100	500	6000	60	0%	0.562
11	Four	10	1000	6000	60	0%	0.562
12	Four	10	500	6000	60	0%	0.562

Table 10 is sorted by resulting total power system mass. The data represent four distinct families of possibilities based on inputs that yield matching results. As shown in Table 9, the average mission power is 61 W. For reference, 93% and 122% of this value is 57 W and 74 W, respectively.

- *Result Family One*: Minimum power system mass occurs when the EDS is sized to 93% of the mission average power. The PDS is required to provide more energy than it is capable of providing, and relies on the EDS for energy. However, for the EDS to share energy, it must supply a certain level of power. The minimum value of 500 W/kg is adequate. When EDS_{SE} of 1000 W-hr/kg is selected, the MAV should be capable of longer endurance. When compared to the other cases, it is shown that 500 W/kg is enough, but 100 W/kg is not sufficient. This suggests that a best value may exist between these design points.

- *Result Family Two*: The next best case uses an EDS sized to 122% of the mission segment average. Due to the low value of PDS energy, it must rely on the EDS. This situation is very similar to Case 1. The EDS is capable of providing high enough power to make it worthwhile to use. The EDS must contribute a certain amount of power in order to contribute energy. The only reason this is viable is because the EDS_{SP} is quite high. As in *Result Family One*, EDS_{SE} of 500 W-hr/kg is adequate for this mission, but the 1000 W-hr/kg EDS_{SE} would otherwise provide longer endurance.
- *Result Family Three*: A more realistic scenario is illustrated here. Runs 5-8 agree that the Li-Po PDS is good enough to complete the mission with no EDS hybridization. The shaded region refers to values not applicable because it is a PDS system only. The mass is driven up because the PDS must meet the energy requirement.
- *Result Family Four*: These results are similar to *Result Family Three*. No degree of hybridization is better than having the Li-Ion battery alone, as shown by runs 9-12. The shaded region refers to values not applicable because it is a PDS system only. The mass calculation is driven by meeting the energy requirement. The PDS_{SE} is not as good as in *Result Family Three*, so the system weight is greater.

IV.4.5. Analysis 4: EDS Values for Hour Long Mission

The final set of analysis seeks to determine the power source composition for a lengthened mission, as compared to the shorter mission. First, inputs from section IV.4.3 are used to simulate an hour-long mission. Then, in section IV.4.5.2, Li-Po batteries are hybridized with a small fuel cell over a longer mission. Lastly, in section IV.4.5.3, EDS

values are determined for Li-Po and Li-Ion hybridization that will be required to reduce the power system mass as compared to a single Li-Po battery.

IV.4.5.1. Extending the Mission

It has been shown that for a specified 30 min mission using a hybridized power source made up of a large fuel cell and a current-technology Li-Po battery, a reduced mass can be achieved by hybridizing. In the specific case analyzed, the EDS should be sized at approximately 90% power of the mission average power. When the same type of mission is extended from 30 to 60 min, energy requirements will increase the power system mass. The new degree of hybridization will be explored. At this point, however, the mass increase will not exceed 125% of the original MAV mass, which has been shown to be flyable, but new mission power requirements driven from increased mass have not been determined.

Ten additional minutes were added to each of the middle three mission segments shown in Figure 11 to create a 60 min mission with the same speeds, climb, and descend rates. The five design variables are optimized to minimize mass with bounds given in Table 11. This includes an optimistic EDS, and a PDS with values of today's Li-Po batteries.

Table 11. Design space for large fuel cell and Li-Po battery.

Design Variable	Units	Minimum	Maximum	Representation
EDS_{SP}	W/kg	10	500	Small to large FC
EDS_{SE}	W-hr/kg	500	1000	Small to large FC
PDS_{SP}	W/kg	0	1200	Current Li-Po
PDS_{SE}	W-hr/kg	0	160	Current Li-Po
Differential	Percent	0	133	No hybrid to full hybrid

The results for the 30 and 60 min runs are shown in Table 12. The optimized component values are shown in Table 13, where highlighted numbers indicate active constraints. The hour long, optimized split power profile is shown in Figure 67.

Table 12. Effect on differential due to mission duration.

	Units	30 Min Mission	60 Min Mission
Maximum PDS_{SE}	W-hr/kg	160	160
Mission power max	W	86	86
Mission power average	W	61	62
Mission energy	W-hr	33	73
Differential	%	90	115
PDS + EDS mass	g	143	155
Total power system mass	g	157	170

Table 13. Optimized component values for 30 and 60 min mission.

Design Variable	Units	30 Min Mission	60 Min Mission
EDS_{SP}	W/kg	500	500
EDS_{SE}	W-hr/kg	669	721
PDS_{SP}	W/kg	939	1200
PDS_{SE}	W-hr/kg	160	160
Differential	Percent	90	115

Table 12 shows that by doubling the duration of the mission, more energy was required. It can be seen in Table 13 that both energy and power were active constraints for the PDS. In both mission durations, the EDS_{SP} and PDS_{SE} were used to the full potential. As the energy requirement increased, so did the amount of hybridization. The relative size of the EDS increased as expected. It went from 90% to 115%, which indicates that for this case, the EDS should be sized for slightly less than average mission

power for the shorter mission, and slightly more than average mission power for the longer mission.

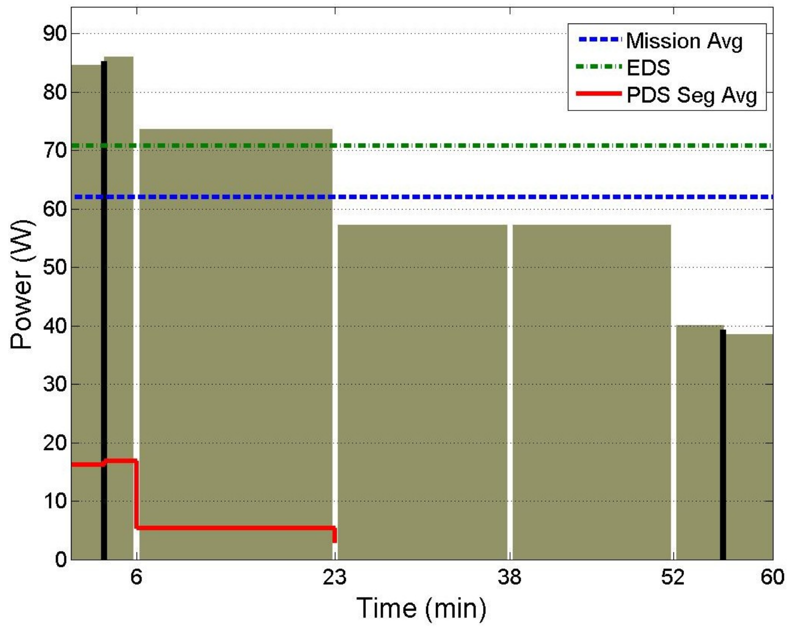

Figure 67. Optimized split and power profile for hour-long mission.

The previous results came from using very high values for a small fuel cell that are yet unrealistic for a MAV. The next analysis reduces the EDS values to make conclusions that are more relevant.

IV.4.5.2. Small DMFC and Li-Po

Here, the framework is applied to a more realistic situation. Assuming that the DMFC identified in section II.4.2.4 currently being tested for soldier portable power could provide an ambitious 25 W/kg and is volumetrically compatible with the GenMAV, this DMFC is now used to try to hybridize with a current Li-Po battery. Results from the half-hour and hour-long mission are compared in Table 14.

Table 14. DMFC and Li-Po power system hybrid results.

Design Variable	Units	30 min	60 min
EDS_{SP}	W/kg	25	25
EDS_{SE}	W-hr/kg	550	550
PDS_{SP}	W/kg	1200	1200
PDS_{SE}	W-hr/kg	160	160
Differential	Percent	0	0
Total Mass (battery)	g	192	388

Both mission durations failed to utilize the DMFC EDS to save mass. Total mass for the 60 min mission is slightly more than twice the 30 min mission because a small increase in average power. Comparing these battery-only results with the hybrid-power results shown in Table 12, for these circumstances, a hybridization scheme could reduce power system mass by 18% and 56% for the respective 30 and 60 min missions, if a hypothetical 500 W/kg MAV fuel cell were available. However, this analysis shows that given the optimistic fuel cell component combined with a current Li-Po battery, a mass reduction cannot yet be achieved by hybridization at the MAV scale.

IV.4.5.3. What EDS Values Are Necessary?

Given values for Li-Po and Li-Ion batteries, this section seeks to determine what the EDS capabilities need to be in order to compete with the best attainable battery-only option. The EDS_{SP} was initially bounded from 25-500 W/kg, and EDS_{SE} from 550-1000 W-hr/kg. The lower bounds represent assumptions of the DMFC previously discussed. The higher bounds are unrealistic for a small fuel cell, but possible for a much larger fuel cell system. Model Center's optimization was used to solve for a total power system mass slightly less than the best previously found. For the half-hour mission, the power system mass with 10% packaging factor was found to be 210 g. In order to find the

threshold values that would signal minimum EDS capability requirements for a hybrid system of reduced mass, the optimization routine solved for 200 g. The 10 g difference was used to ensure that optimization tolerances did not prematurely stop the search. For the full-hour mission, the Li-Po only solution was 426 g, so the optimization routine searched for a total power mass goal of 420 g to signal a threshold value for an EDS. Note that these Li-Po solutions are the same as shown in Table 14, but include the packaging factor. Should the resulting differential value be greater than zero, it signals minimum values for an EDS. Realistically, however, the small benefit provided would not outweigh the complexity of adding a fuel cell at at such small fractions. Differential values would likely need to be closer to 80% in order for such system to be practical. For this analysis and all subsequent analyses, results are feasible solutions, but not necessarily optimal solutions. This is because results came from solving for a certain mass, indicative of a threshold, instead of solving for the minimum value. Values for the EDS and the differential were shown in Table 15.

Table 15. Possible hybridization with battery and future EDS.

	PDS_{SP} [W/kg]	PDS_{SE} [W-hr/kg]	EDS_{SP} [W/kg]	EDS_{SE} [W-hr/kg]	Diff [%]
Li-Po, 30 min	1200	160	325	921	58
Li-Ion, 30 min	6000	60	468	653	113
Li-Po, 60 min	1200	160	470	989	2
Li-Ion, 60 min	6000	60	358	785	78

Both the shorter and longer missions provide an opportunity for a good EDS matched with a Li-Po to reduce the power system mass when compared to only a Li-Po battery power system. Since the Li-Ion provides low energy compared to the Li-Po and

the EDS, a hybrid Li-Ion/EDS system can also reduce power system mass as compared to a Li-Po battery only. The comparison is made to the Li-Po because it performs better and results in a reduced mass for these missions as compared to the Li-Ion. A fuel cell comprised of only 2% of the average mission power would only need to provide 1 W. The added cost and complexity would likely limit its inclusion within a power system, but these results show that as these systems become more economical, performance can be enhanced by including them, even at low power. Because the EDS changes between the short and long mission, the differential values are not directly comparable.

In order to further determine fuel cell property values suitable for Li-Po hybridization for the subject MAV, the EDS_{SP} and EDS_{SE} values were simultaneously increased at 50% increments from the baseline DMFC. For example, limits on EDS_{SP} were set to 25 W/kg, 37.5 W/kg, 50 W/kg, etc. EDS_{SE} increased in the pattern of 550 W-hr/kg, 825 W-hr/kg, 1100 W-hr/kg, etc. Even up to 1500 W-hr/kg, which represented a very high value for a small fuel cell, no feasible solution was found. The value of EDS_{SE} was bounded at this point. The EDS_{SP} was incrementally increased to 13 times the subject DMFC value before a feasible design was found. For comparison, the full-hour results shown in Table 13 are repeated with the half-hour results in Table 16.

Table 16. EDS goal for hybridization.

	PDS_{SP} [W/kg]	PDS_{SE} [W-hr/kg]	EDS_{SP} [W/kg]	EDS_{SE} [W-hr/kg]	Diff [%]
30 min mission, Li-Po	1200	160	325	921	58
60 min mission, Li-Po	1200	160	500	721	115

The EDS values for the 30 min mission represent a feasible solution found by incrementing the EDS bounds, and using an optimization technique to solve for values

that would produce a power system mass less than that required of today's battery-only system. This may be far from optimal, but identifies values that could produce a reduced mass system through hybridization. The full-hour results did not use an incremental technique, but instead minimized mass with generous EDS bounds. Through not directly comparable, the longer mission had greater energy requirements, as expected. Table 16 illustrates that with an EDS capable of at least 325 W/kg and 921 W-hr/kg, weight savings through hybridization with a current Li-Po battery can be achieved for a half-hour mission. The higher differential for the hour-long mission indicates that hybridization would be even more beneficial for a longer mission.

The conceptual design tool simulated various energy and power specific values of a future EDS scaled up from a DMFC. In order to be viable for the conditions in these simulations, the fuel cell must increase its specific power by 1200%. Feasible future EDS values were identified that when hybridized with a current Li-Po battery, could reduce the power system mass. These results indicate the EDS must improve capability to provide at least 325 W/kg matched with at least 921 W-hr/kg to be part of a hybrid solution better than Li-Po batteries alone.

The final step in a power system design would then update the power system mass within the MAV, recalculate the mission power profile and iterate until some prescribed tolerance is met. Complete and capable system-level fuel cells (with fuel) do not presently exist at these small masses, and if they did, they still may be volumetrically prohibitive. However, results presented here help identify reasonable goals for fuel cell technology.

V. Conclusions of Research

V.1. Conclusions

This effort was successful on several fronts. Most importantly, it set the stage for future work. All of the background research regarding low Reynolds number propeller and wing aerodynamics, small propellers, low aspect ratio planforms, electric power plants, fuel cell BOP integration, and scalable power systems must be integrated in order to further reduce the sizes of useful MAVs. This conceptual design tool provides a framework for further multi-disciplinary collaboration.

In response to the sponsor's need for power system optimization, this tool analyzed a power system based on a set airframe, although iterations on MAV design could easily be performed with only minor changes. It was felt that a greater benefit could be had through studying near-term integration opportunities, rather than farther off feasibility studies, so this tool was tailored to provide realistic information and answer questions about what can be done with technology as it becomes available today.

Hybrid-power systems were analyzed within the Model Center framework for a simulated GenMAV mission. A range of possible power sources were matched and compared with requirements derived from a single mission profile. Due to the current technology of small fuel cells with low specific power, results show that a MAV-sized fuel cell-battery hybrid-power system would not perform better than a pure battery or battery-battery power system. However, a feasible fuel cell capable of providing at least 325 W/kg matched with at least 921 W-hr/kg was identified as a Li-Po – fuel cell solution that would reduce power system mass compared to using only Li-Po batteries. Utilizing

enhanced fuel cell specific power and energy properties and current Li-Po battery properties, both a 30 min and 60 min mission were shown to have a reduced mass through hybridization. Results indicate that if the EDS has sufficient specific power to compete with batteries, weight savings can be achieved via EDS/PDS hybridization. Specific power of the EDS is the technological bottleneck. The feasibility and sizing strategy of hybrid-power components depends upon both mission requirements and power source component intrinsic properties. Without a statistically driven analysis across various mission profiles, one cannot be certain these results apply to other types of mission profiles. It must be assumed that the resulting numbers found in this effort are highly dependent on the mission-driven power and energy requirements. It is clear, however, that once EDS specific power increases, hybridization will lead to overall reduced weight.

This effort had several outcomes and general conclusions. The following list highlights some capability of the conceptual design framework and general conclusions.

1. A tool has been built that will allow a user to quickly assess emerging power technologies for MAV and UAS applications.
2. This tool can be used for PDS-EDS (such as battery verses a fuel cell) hybrid-power combinations or more general PDS-PDS (PDS Li-Ion battery verses relative EDS Li-Po battery) hybrid-power combinations.
3. Trade studies can be performed on the GenMAV-based model for improved selection of components such as motors and propellers. Geometric scaling could be evaluated. Capability and flight quality could be analyzed with a changing power system mass.

4. Aerodynamic stability and control effects are readily visible through the AVL model within Model Center. They can be used to judge performance implications of any changes made to the flight vehicle.
5. AVL and QPROP are good, easy to use, computationally cheap tools very applicable to conceptual design.
6. Model Center provides an excellent framework for multi-disciplinary integration. This allows a single user or even a team of users to perform work that used to require multiple groups of personnel.
7. No a priori sizing criterion exists for fuel cells. Fuel cells, or any power source for that matter, should be sized for mission requirements and matched with power source intensive properties.
8. Lengthening the mission increases energy requirements. Analysis performed for this work resulted in a trend related to the degree of hybridization: once a mission and configuration is found to benefit from hybridization, the degree of hybridization increases in proportion to the required endurance. More simulation and validation will be required to confirm this preliminary conclusion for a variety of mission lengths over a wide flight envelope.
9. Comparing the results of a hybrid configuration that used EDS values not yet technologically available with a current-technology battery-only configuration, a specific example showed that a hybridization scheme could reduce power system mass by 18% and 56% for the respective 30 and 60 min missions.
10. Small fuel cells must substantially improve specific power capability to compete with improving battery chemistries. A fuel cell-battery hybrid-power configuration

cannot yet achieve a mass reduction compared to a battery-only option at the MAV scale

V.2. Model Shortfalls

Every endeavor is limited by time. Some of the *Shortfalls* of this research have been postponed, and will be included in the *Recommendations* section. Others, such as those listed below, might have improved the model's robustness and increased validity of the results.

1. The power management model has no way of utilizing less than maximum specific power and maximum specific energy, even though both do not occur simultaneously. Perhaps the easiest way to fix this is to calculate everything based on polarization curves or Ragone plots from sample test data, which could be the baseline for an unknown power source component.
2. At certain times during this effort's model design, a feedback loop was incorporated to adjust the baseline GenMav model for both changing altitude and changing MAV mass due to hydrogen fuel burn. This was determined to be a relatively insignificant mass change; therefore, it was omitted. Reincorporating a fuel burn mass change component may be useful for airframe design, especially for some types of fuel cells that may actually gain mass during the mission.
3. The "Power Plot," is based on multiple overlaid bar and scatter plots. An example of this plot is Figure 67. Despite much tenacity, the plot window of the bar plots and scatter plots do not share the same exact scale. The calculations are all correct

and the analysis is correct, but the plot displayed will show the lines to be a few Watts higher than they should be as compared to the bars.

4. The most important aspect of efficiency is matching both the motor and propeller with the electrical power. The propeller used in this study is simply a smaller propeller with a different pitch. It has been geometrically stretched, and may not represent a real propeller. It certainly does not represent the propeller used on the GenMAV. A better propeller model for QPROP is required.

V.3. Recommendations for Future Research

Like any major endeavor, this research effort was subject to quite a bit of evolution. It was not clear in the beginning if a MAV should be built or just simulated, and for what purpose. Only after performing some literature review, becoming intrigued with fuel cells, and meeting with the sponsor did a clear need become apparent. A multi-disciplinary design tool was built to answer practical questions for real systems. However, it also laid the groundwork for future fuel cell-powered MAV application. More focused and in-depth models should be added to the tool to improve the accuracy, ease of use, and speed. The following items are some improvements that would extend the utility and validity of this effort.

1. Obtain the actual propeller used in the GenMAV and model it according to QPROP instructions. The length, pitch, and chord as a function of radius could be measured easily enough in order to build the geometry model. For the aerodynamic model, the simplest path would be to assume the propeller airfoil does not change radially, and select a NACA airfoil that has available aerodynamic data.

Or, a better option would be to draft the airfoil geometry to a file that describes the shape and use XFOIL to determine the aerodynamic characteristics. Only then could results actually represent the GenMAV.

2. The conceptual design framework is currently set up to utilized endurance as a user input. It may be useful to modify the program to make endurance an output, based on the type of power components used. This would be a relatively simple change to the “energy.m” file under the Post Processing component, and the variable types within Model Center. Otherwise, endurance information can be found with the present model through iteration.
3. Incorporate a feedback loop to recalculate the total electric power required based on an optimized hybrid source power system mass. Iterate on the loop until the weight of the power system has been completely incorporated into the mission power-required determination.
4. Follow up on the work of Beam, et al. in order to make this multidisciplinary design tool internet based using Model Center for quick plug and play model capability.[86] This was discussed in II.7. This may not be especially valuable given the difficulty of connecting through firewalled networks, but a cyber-minded individual could then enhance collaboration opportunities.
5. Incorporate the fuel cell model described in section III.2.5. A more detailed polarization model utilizing the same principles as above that accounts for reactant partial pressure, temperature, membrane water content, and current density can be found in reference 89.

6. Incorporate the battery model described in section III.2.6. Include SOC methodology and in-flight charging capability. Once time-dependent simulations are included within the model, capacitors can be incorporated with the use of SOC indicators and duty cycle electronics.
7. Improve the power management scheme. This would be a model to get the proper voltage and current from the source to the sink and integrate the various sources. This may include designing an electrical controller that accounts for the efficiency losses and dynamic nature of requirements and supply availability.
8. Once electrical control is better understood, the power system for a flapping wing model could be integrated more easily.
9. Collaborate better with electrical and computer science engineering specialists. This effort was lead by a student of aerodynamic engineering with a mechanical engineering background, and therefore suffered from a lack of in-depth software authoring and electrical system design experience. If a group of students with different backgrounds collaborated on this effort, the potential exists for significant gain in utility.
10. Many tangent paths are listed within Mustafa Turan's thesis (Ref. 88) as known model shortfalls, but it is up to our imagination and interest to focus on the most valuable paths. Topics include any specialty of aircraft design (i.e. stability and control, structures, aerodynamics, and propulsion).
11. Employ optimal control tools such as dynamic programming to optimize mission segments, propulsion components, power system components, and other design variables.

Appendix A: Known Software Issues and Tips

The following is a short list of challenges experienced working with Model Center. Some of them are associated with how the models interact, and others are software bugs that should be fixed in the next release of Model Center.

1. In performing AVL-Model Center verification, certain outputs were not changing with changed inputs. Within Model Center, component background outputs can be viewed through Component-"Progress Stdout". It was found that the order of flight condition input into AVL affected the results. Whenever a flight condition variable is changed, a new lift coefficient is calculated for equilibrium flight. The instructions for Model Center are contained in "constraints_GenMAV.batch." When density was changed first, followed by velocity, and then mass, AVL assumes that the current C_L is max C_L, and then provides a new velocity different from the requested velocity. To work around this, "constraint_GenMAV.batch" was modified so that mass is changed first, followed by density, and velocity last. This solved the problem and Model Center inputs now reflect the actual value used in AVL. This is no longer an issue, but familiarity may help with other issues.
2. When the optimization GUI is closed by closing the dialog window, Model Center asks to be restarted, and often crashes. If "OK" is pressed, it runs the optimization again. The only work-around is to depress "OK" and let the optimization start. Then depress "Halt," and the dialog box will close. Hopefully Model Center version 9.0 fixed this bug.

3. The “Run” button can sometimes hang, and not provide an immediate response. Be patient; the model should start running. If it hasn’t started in 10 seconds, it is generally safe to press it again. If “Run” is pressed twice, it will run twice in a row.
4. Before running Script Scheduler, must always … Tools, Scheduler, Reset all as invalid. When this is done, all components will be run, which prevents the passing of stale data. This should not be a problem, because the model has been completely rewritten as to not use the script scheduler.
5. Sometimes, especially after multiple changes are done, it is necessary to reset Model Center. This model can cause multiple instances of Matlab to run in the background. This utilizes many resources, and after some time, the computer RAM and page file will become completely utilized. There is no clean way to close the Matlab instances other than restarting Model Center and Analysis Server.
6. When running parametric studies, ensure that the box for “Validate All” is left unchecked. Also ensure that the “Mission Driver” is not set on “auto run.”
7. Input variables passed to multiple components should be done in parallel. If inputs are passed through components that don’t change the variable, Model Center sometimes doesn’t know the order in which to run the model.
8. The 3-D toolbox in the Data Visualization application is very resource intensive. Save work often because it has been known to crash when hovering over a data point.

Appendix B: Software Versions

The conceptual design framework built for this research included many components and various software. Below is a listing of the most important software used along with the version number used.

- Model Center Version 8.07
    - Optimization\Design Explorer Version: 2.2.5
    - Optimization\Gradient Optimizer Version: 2.2.5
    - Converger Version: 1.2.0
    - Design Explorer Version: 2.2.5
    - Excel Plug-In Version: 1.4.4
    - Gradient Optimizer Version: 2.2.5
    - Matlab Plug-In Version: 1.6.5
- Analysis Server v.6.01 Build 29479
- Mathworks MatLab R2009b
- Microsoft Excel 2003, SP3
- All programming and analysis was run on an IBM R51 Thinkpad laptop with an Intel Pentium M processor (1500MHz) with 2 GB RAM
- Microsoft Windows XP Professional, Service Pack 3
- Athena Vortex Lattice (AVL) Version 3.26
- QPROP Version 1.21

Appendix C: Design of Experiments Results

A DOE was presented in section IV.4.2. The following table lists the result variables tracked while performing the 3-factorial DOE. The input values were suggested by the research sponsor, and they are presented here by request. For the simulated mission, the average power was 61.17 W, and the peak power was 86.03 W.

EDS SP W/kg	EDS SE W-hr/kg	PDS SP W/kg	PDS SE W-hr/kg	diff %	Pwr EDS W	Engy EDS W-hr	mass EDS kg	Pwr PDS W	Engy PDS W-hr	mass PDS kg	PDS time hr	Tot. mass kg
10	500	1200	60	0%	0	0	0	86.03	30.64	0.511	0.501	0.562
255	500	1200	60	0%	0	0	0	86.03	30.64	0.511	0.501	0.562
500	500	1200	60	0%	0	0	0	86.03	30.64	0.511	0.501	0.562
10	750	1200	60	0%	0	0	0	86.03	30.64	0.511	0.501	0.562
255	750	1200	60	0%	0	0	0	86.03	30.64	0.511	0.501	0.562
500	750	1200	60	0%	0	0	0	86.03	30.64	0.511	0.501	0.562
10	1000	1200	60	0%	0	0	0	86.03	30.64	0.511	0.501	0.562
255	1000	1200	60	0%	0	0	0	86.03	30.64	0.511	0.501	0.562
500	1000	1200	60	0%	0	0	0	86.03	30.64	0.511	0.501	0.562
10	500	3600	60	0%	0	0	0	86.03	30.64	0.511	0.501	0.562
255	500	3600	60	0%	0	0	0	86.03	30.64	0.511	0.501	0.562
500	500	3600	60	0%	0	0	0	86.03	30.64	0.511	0.501	0.562
10	750	3600	60	0%	0	0	0	86.03	30.64	0.511	0.501	0.562
255	750	3600	60	0%	0	0	0	86.03	30.64	0.511	0.501	0.562
500	750	3600	60	0%	0	0	0	86.03	30.64	0.511	0.501	0.562
10	1000	3600	60	0%	0	0	0	86.03	30.64	0.511	0.501	0.562
255	1000	3600	60	0%	0	0	0	86.03	30.64	0.511	0.501	0.562
500	1000	3600	60	0%	0	0	0	86.03	30.64	0.511	0.501	0.562
10	500	6000	60	0%	0	0	0	86.03	30.64	0.511	0.501	0.562
255	500	6000	60	0%	0	0	0	86.03	30.64	0.511	0.501	0.562
500	500	6000	60	0%	0	0	0	86.03	30.64	0.511	0.501	0.562
10	750	6000	60	0%	0	0	0	86.03	30.64	0.511	0.501	0.562
255	750	6000	60	0%	0	0	0	86.03	30.64	0.511	0.501	0.562
500	750	6000	60	0%	0	0	0	86.03	30.64	0.511	0.501	0.562
10	1000	6000	60	0%	0	0	0	86.03	30.64	0.511	0.501	0.562
255	1000	6000	60	0%	0	0	0	86.03	30.64	0.511	0.501	0.562
500	1000	6000	60	0%	0	0	0	86.03	30.64	0.511	0.501	0.562
10	500	1200	130	0%	0	0	0	86.03	30.64	0.236	0.501	0.259
255	500	1200	130	0%	0	0	0	86.03	30.64	0.236	0.501	0.259
500	500	1200	130	0%	0	0	0	86.03	30.64	0.236	0.501	0.259

EDS SP W/kg	EDS SE W-hr/kg	PDS SP W/kg	PDS SE W-hr/kg	diff %	Pwr EDS W	Engy EDS W-hr	mass EDS kg	Pwr PDS W	Engy PDS W-hr	mass PDS kg	PDS time hr	Tot. mass kg
10	750	1200	130	0%	0	0	0	86.03	30.64	0.236	0.501	0.259
255	750	1200	130	0%	0	0	0	86.03	30.64	0.236	0.501	0.259
500	750	1200	130	0%	0	0	0	86.03	30.64	0.236	0.501	0.259
10	1000	1200	130	0%	0	0	0	86.03	30.64	0.236	0.501	0.259
255	1000	1200	130	0%	0	0	0	86.03	30.64	0.236	0.501	0.259
500	1000	1200	130	0%	0	0	0	86.03	30.64	0.236	0.501	0.259
10	500	3600	130	0%	0	0	0	86.03	30.64	0.236	0.501	0.259
255	500	3600	130	0%	0	0	0	86.03	30.64	0.236	0.501	0.259
500	500	3600	130	0%	0	0	0	86.03	30.64	0.236	0.501	0.259
10	750	3600	130	0%	0	0	0	86.03	30.64	0.236	0.501	0.259
255	750	3600	130	0%	0	0	0	86.03	30.64	0.236	0.501	0.259
500	750	3600	130	0%	0	0	0	86.03	30.64	0.236	0.501	0.259
10	1000	3600	130	0%	0	0	0	86.03	30.64	0.236	0.501	0.259
255	1000	3600	130	0%	0	0	0	86.03	30.64	0.236	0.501	0.259
500	1000	3600	130	0%	0	0	0	86.03	30.64	0.236	0.501	0.259
10	500	6000	130	0%	0	0	0	86.03	30.64	0.236	0.501	0.259
255	500	6000	130	0%	0	0	0	86.03	30.64	0.236	0.501	0.259
500	500	6000	130	0%	0	0	0	86.03	30.64	0.236	0.501	0.259
10	750	6000	130	0%	0	0	0	86.03	30.64	0.236	0.501	0.259
255	750	6000	130	0%	0	0	0	86.03	30.64	0.236	0.501	0.259
500	750	6000	130	0%	0	0	0	86.03	30.64	0.236	0.501	0.259
10	1000	6000	130	0%	0	0	0	86.03	30.64	0.236	0.501	0.259
255	1000	6000	130	0%	0	0	0	86.03	30.64	0.236	0.501	0.259
500	1000	6000	130	0%	0	0	0	86.03	30.64	0.236	0.501	0.259
10	500	1200	200	0%	0	0	0	86.03	30.64	0.153	0.501	0.169
255	500	1200	200	0%	0	0	0	86.03	30.64	0.153	0.501	0.169
500	500	1200	200	0%	0	0	0	86.03	30.64	0.153	0.501	0.169
10	750	1200	200	0%	0	0	0	86.03	30.64	0.153	0.501	0.169
255	750	1200	200	0%	0	0	0	86.03	30.64	0.153	0.501	0.169
500	750	1200	200	0%	0	0	0	86.03	30.64	0.153	0.501	0.169
10	1000	1200	200	0%	0	0	0	86.03	30.64	0.153	0.501	0.169
255	1000	1200	200	0%	0	0	0	86.03	30.64	0.153	0.501	0.169
500	1000	1200	200	0%	0	0	0	86.03	30.64	0.153	0.501	0.169
10	500	3600	200	0%	0	0	0	86.03	30.64	0.153	0.501	0.169
255	500	3600	200	0%	0	0	0	86.03	30.64	0.153	0.501	0.169
500	500	3600	200	0%	0	0	0	86.03	30.64	0.153	0.501	0.169
10	750	3600	200	0%	0	0	0	86.03	30.64	0.153	0.501	0.169
255	750	3600	200	0%	0	0	0	86.03	30.64	0.153	0.501	0.169
500	750	3600	200	0%	0	0	0	86.03	30.64	0.153	0.501	0.169
10	1000	3600	200	0%	0	0	0	86.03	30.64	0.153	0.501	0.169
255	1000	3600	200	0%	0	0	0	86.03	30.64	0.153	0.501	0.169
500	1000	3600	200	0%	0	0	0	86.03	30.64	0.153	0.501	0.169
10	500	6000	200	0%	0	0	0	86.03	30.64	0.153	0.501	0.169

EDS SP W/kg	EDS SE W-hr/kg	PDS SP W/kg	PDS SE W-hr/kg	diff %	Pwr EDS W	Engy EDS W-hr	mass EDS kg	Pwr PDS W	Engy PDS W-hr	mass PDS kg	PDS time hr	Tot. mass kg
255	500	6000	200	0%	0	0	0	86.03	30.64	0.153	0.501	0.169
500	500	6000	200	0%	0	0	0	86.03	30.64	0.153	0.501	0.169
10	750	6000	200	0%	0	0	0	86.03	30.64	0.153	0.501	0.169
255	750	6000	200	0%	0	0	0	86.03	30.64	0.153	0.501	0.169
500	750	6000	200	0%	0	0	0	86.03	30.64	0.153	0.501	0.169
10	1000	6000	200	0%	0	0	0	86.03	30.64	0.153	0.501	0.169
255	1000	6000	200	0%	0	0	0	86.03	30.64	0.153	0.501	0.169
500	1000	6000	200	0%	0	0	0	86.03	30.64	0.153	0.501	0.169
10	500	1200	60	67%	40.68	20.378	4.0677	45.35	10.46	0.174	0.359	4.666
255	500	1200	60	67%	40.68	20.378	0.1595	45.35	10.46	0.174	0.359	0.367
500	500	1200	60	67%	40.68	20.378	0.0814	45.35	10.46	0.174	0.359	0.281
10	750	1200	60	67%	40.68	20.378	4.0677	45.35	10.46	0.174	0.359	4.666
255	750	1200	60	67%	40.68	20.378	0.1595	45.35	10.46	0.174	0.359	0.367
500	750	1200	60	67%	40.68	20.378	0.0814	45.35	10.46	0.174	0.359	0.281
10	1000	1200	60	67%	40.68	20.378	4.0677	45.35	10.46	0.174	0.359	4.666
255	1000	1200	60	67%	40.68	20.378	0.1595	45.35	10.46	0.174	0.359	0.367
500	1000	1200	60	67%	40.68	20.378	0.0814	45.35	10.46	0.174	0.359	0.281
10	500	3600	60	67%	40.68	20.378	4.0677	45.35	10.46	0.174	0.359	4.666
255	500	3600	60	67%	40.68	20.378	0.1595	45.35	10.46	0.174	0.359	0.367
500	500	3600	60	67%	40.68	20.378	0.0814	45.35	10.46	0.174	0.359	0.281
10	750	3600	60	67%	40.68	20.378	4.0677	45.35	10.46	0.174	0.359	4.666
255	750	3600	60	67%	40.68	20.378	0.1595	45.35	10.46	0.174	0.359	0.367
500	750	3600	60	67%	40.68	20.378	0.0814	45.35	10.46	0.174	0.359	0.281
10	1000	3600	60	67%	40.68	20.378	4.0677	45.35	10.46	0.174	0.359	4.666
255	1000	3600	60	67%	40.68	20.378	0.1595	45.35	10.46	0.174	0.359	0.367
500	1000	3600	60	67%	40.68	20.378	0.0814	45.35	10.46	0.174	0.359	0.281
10	500	6000	60	67%	40.68	20.378	4.0677	45.35	10.46	0.174	0.359	4.666
255	500	6000	60	67%	40.68	20.378	0.1595	45.35	10.46	0.174	0.359	0.367
500	500	6000	60	67%	40.68	20.378	0.0814	45.35	10.46	0.174	0.359	0.281
10	750	6000	60	67%	40.68	20.378	4.0677	45.35	10.46	0.174	0.359	4.666
255	750	6000	60	67%	40.68	20.378	0.1595	45.35	10.46	0.174	0.359	0.367
500	750	6000	60	67%	40.68	20.378	0.0814	45.35	10.46	0.174	0.359	0.281
10	1000	6000	60	67%	40.68	20.378	4.0677	45.35	10.46	0.174	0.359	4.666
255	1000	6000	60	67%	40.68	20.378	0.1595	45.35	10.46	0.174	0.359	0.367
500	1000	6000	60	67%	40.68	20.378	0.0814	45.35	10.46	0.174	0.359	0.281
10	500	1200	130	67%	40.68	20.378	4.0677	45.35	10.46	0.080	0.359	4.563
255	500	1200	130	67%	40.68	20.378	0.1595	45.35	10.46	0.080	0.359	0.264
500	500	1200	130	67%	40.68	20.378	0.0814	45.35	10.46	0.080	0.359	0.178
10	750	1200	130	67%	40.68	20.378	4.0677	45.35	10.46	0.080	0.359	4.563
255	750	1200	130	67%	40.68	20.378	0.1595	45.35	10.46	0.080	0.359	0.264
500	750	1200	130	67%	40.68	20.378	0.0814	45.35	10.46	0.080	0.359	0.178
10	1000	1200	130	67%	40.68	20.378	4.0677	45.35	10.46	0.080	0.359	4.563
255	1000	1200	130	67%	40.68	20.378	0.1595	45.35	10.46	0.080	0.359	0.264

EDS SP W/kg	EDS SE W-hr/kg	PDS SP W/kg	PDS SE W-hr/kg	diff %	Pwr EDS W	Engy EDS W-hr	mass EDS kg	Pwr PDS W	Engy PDS W-hr	mass PDS kg	PDS time hr	Tot. mass kg
500	1000	1200	130	67%	40.68	20.378	0.0814	45.35	10.46	0.080	0.359	0.178
10	500	3600	130	67%	40.68	20.378	4.0677	45.35	10.46	0.080	0.359	4.563
255	500	3600	130	67%	40.68	20.378	0.1595	45.35	10.46	0.080	0.359	0.264
500	500	3600	130	67%	40.68	20.378	0.0814	45.35	10.46	0.080	0.359	0.178
10	750	3600	130	67%	40.68	20.378	4.0677	45.35	10.46	0.080	0.359	4.563
255	750	3600	130	67%	40.68	20.378	0.1595	45.35	10.46	0.080	0.359	0.264
500	750	3600	130	67%	40.68	20.378	0.0814	45.35	10.46	0.080	0.359	0.178
10	1000	3600	130	67%	40.68	20.378	4.0677	45.35	10.46	0.080	0.359	4.563
255	1000	3600	130	67%	40.68	20.378	0.1595	45.35	10.46	0.080	0.359	0.264
500	1000	3600	130	67%	40.68	20.378	0.0814	45.35	10.46	0.080	0.359	0.178
10	500	6000	130	67%	40.68	20.378	4.0677	45.35	10.46	0.080	0.359	4.563
255	500	6000	130	67%	40.68	20.378	0.1595	45.35	10.46	0.080	0.359	0.264
500	500	6000	130	67%	40.68	20.378	0.0814	45.35	10.46	0.080	0.359	0.178
10	750	6000	130	67%	40.68	20.378	4.0677	45.35	10.46	0.080	0.359	4.563
255	750	6000	130	67%	40.68	20.378	0.1595	45.35	10.46	0.080	0.359	0.264
500	750	6000	130	67%	40.68	20.378	0.0814	45.35	10.46	0.080	0.359	0.178
10	1000	6000	130	67%	40.68	20.378	4.0677	45.35	10.46	0.080	0.359	4.563
255	1000	6000	130	67%	40.68	20.378	0.1595	45.35	10.46	0.080	0.359	0.264
500	1000	6000	130	67%	40.68	20.378	0.0814	45.35	10.46	0.080	0.359	0.178
10	500	1200	200	67%	40.68	20.378	4.0677	45.35	10.46	0.052	0.359	4.532
255	500	1200	200	67%	40.68	20.378	0.1595	45.35	10.46	0.052	0.359	0.233
500	500	1200	200	67%	40.68	20.378	0.0814	45.35	10.46	0.052	0.359	0.147
10	750	1200	200	67%	40.68	20.378	4.0677	45.35	10.46	0.052	0.359	4.532
255	750	1200	200	67%	40.68	20.378	0.1595	45.35	10.46	0.052	0.359	0.233
500	750	1200	200	67%	40.68	20.378	0.0814	45.35	10.46	0.052	0.359	0.147
10	1000	1200	200	67%	40.68	20.378	4.0677	45.35	10.46	0.052	0.359	4.532
255	1000	1200	200	67%	40.68	20.378	0.1595	45.35	10.46	0.052	0.359	0.233
500	1000	1200	200	67%	40.68	20.378	0.0814	45.35	10.46	0.052	0.359	0.147
10	500	3600	200	67%	40.68	20.378	4.0677	45.35	10.46	0.052	0.359	4.532
255	500	3600	200	67%	40.68	20.378	0.1595	45.35	10.46	0.052	0.359	0.233
500	500	3600	200	67%	40.68	20.378	0.0814	45.35	10.46	0.052	0.359	0.147
10	750	3600	200	67%	40.68	20.378	4.0677	45.35	10.46	0.052	0.359	4.532
255	750	3600	200	67%	40.68	20.378	0.1595	45.35	10.46	0.052	0.359	0.233
500	750	3600	200	67%	40.68	20.378	0.0814	45.35	10.46	0.052	0.359	0.147
10	1000	3600	200	67%	40.68	20.378	4.0677	45.35	10.46	0.052	0.359	4.532
255	1000	3600	200	67%	40.68	20.378	0.1595	45.35	10.46	0.052	0.359	0.233
500	1000	3600	200	67%	40.68	20.378	0.0814	45.35	10.46	0.052	0.359	0.147
10	500	6000	200	67%	40.68	20.378	4.0677	45.35	10.46	0.052	0.359	4.532
255	500	6000	200	67%	40.68	20.378	0.1595	45.35	10.46	0.052	0.359	0.233
500	500	6000	200	67%	40.68	20.378	0.0814	45.35	10.46	0.052	0.359	0.147
10	750	6000	200	67%	40.68	20.378	4.0677	45.35	10.46	0.052	0.359	4.532
255	750	6000	200	67%	40.68	20.378	0.1595	45.35	10.46	0.052	0.359	0.233
500	750	6000	200	67%	40.68	20.378	0.0814	45.35	10.46	0.052	0.359	0.147

EDS SP W/kg	EDS SE W-hr/kg	PDS SP W/kg	PDS SE W-hr/kg	diff %	Pwr EDS W	Engy EDS W-hr	mass EDS kg	Pwr PDS W	Engy PDS W-hr	mass PDS kg	PDS time hr	Tot. mass kg
10	1000	6000	200	67%	40.68	20.378	4.0677	45.35	10.46	0.052	0.359	4.532
255	1000	6000	200	67%	40.68	20.378	0.1595	45.35	10.46	0.052	0.359	0.233
500	1000	6000	200	67%	40.68	20.378	0.0814	45.35	10.46	0.052	0.359	0.147
10	500	1200	60	133%	81.35	40.756	8.1353	4.68	0.37	0.006	0.093	8.956
255	500	1200	60	133%	81.35	40.756	0.319	4.68	0.37	0.006	0.093	0.358
500	500	1200	60	133%	81.35	40.756	0.1627	4.68	0.37	0.006	0.093	0.186
10	750	1200	60	133%	81.35	40.756	8.1353	4.68	0.37	0.006	0.093	8.956
255	750	1200	60	133%	81.35	40.756	0.319	4.68	0.37	0.006	0.093	0.358
500	750	1200	60	133%	81.35	40.756	0.1627	4.68	0.37	0.006	0.093	0.186
10	1000	1200	60	133%	81.35	40.756	8.1353	4.68	0.37	0.006	0.093	8.956
255	1000	1200	60	133%	81.35	40.756	0.319	4.68	0.37	0.006	0.093	0.358
500	1000	1200	60	133%	81.35	40.756	0.1627	4.68	0.37	0.006	0.093	0.186
10	500	3600	60	133%	81.35	40.756	8.1353	4.68	0.37	0.006	0.093	8.956
255	500	3600	60	133%	81.35	40.756	0.319	4.68	0.37	0.006	0.093	0.358
500	500	3600	60	133%	81.35	40.756	0.1627	4.68	0.37	0.006	0.093	0.186
10	750	3600	60	133%	81.35	40.756	8.1353	4.68	0.37	0.006	0.093	8.956
255	750	3600	60	133%	81.35	40.756	0.319	4.68	0.37	0.006	0.093	0.358
500	750	3600	60	133%	81.35	40.756	0.1627	4.68	0.37	0.006	0.093	0.186
10	1000	3600	60	133%	81.35	40.756	8.1353	4.68	0.37	0.006	0.093	8.956
255	1000	3600	60	133%	81.35	40.756	0.319	4.68	0.37	0.006	0.093	0.358
500	1000	3600	60	133%	81.35	40.756	0.1627	4.68	0.37	0.006	0.093	0.186
10	500	6000	60	133%	81.35	40.756	8.1353	4.68	0.37	0.006	0.093	8.956
255	500	6000	60	133%	81.35	40.756	0.319	4.68	0.37	0.006	0.093	0.358
500	500	6000	60	133%	81.35	40.756	0.1627	4.68	0.37	0.006	0.093	0.186
10	750	6000	60	133%	81.35	40.756	8.1353	4.68	0.37	0.006	0.093	8.956
255	750	6000	60	133%	81.35	40.756	0.319	4.68	0.37	0.006	0.093	0.358
500	750	6000	60	133%	81.35	40.756	0.1627	4.68	0.37	0.006	0.093	0.186
10	1000	6000	60	133%	81.35	40.756	8.1353	4.68	0.37	0.006	0.093	8.956
255	1000	6000	60	133%	81.35	40.756	0.319	4.68	0.37	0.006	0.093	0.358
500	1000	6000	60	133%	81.35	40.756	0.1627	4.68	0.37	0.006	0.093	0.186
10	500	1200	130	133%	81.35	40.756	8.1353	4.68	0.37	0.004	0.093	8.953
255	500	1200	130	133%	81.35	40.756	0.319	4.68	0.37	0.004	0.093	0.355
500	500	1200	130	133%	81.35	40.756	0.1627	4.68	0.37	0.004	0.093	0.183
10	750	1200	130	133%	81.35	40.756	8.1353	4.68	0.37	0.004	0.093	8.953
255	750	1200	130	133%	81.35	40.756	0.319	4.68	0.37	0.004	0.093	0.355
500	750	1200	130	133%	81.35	40.756	0.1627	4.68	0.37	0.004	0.093	0.183
10	1000	1200	130	133%	81.35	40.756	8.1353	4.68	0.37	0.004	0.093	8.953
255	1000	1200	130	133%	81.35	40.756	0.319	4.68	0.37	0.004	0.093	0.355
500	1000	1200	130	133%	81.35	40.756	0.1627	4.68	0.37	0.004	0.093	0.183
10	500	3600	130	133%	81.35	40.756	8.1353	4.68	0.37	0.003	0.093	8.952
255	500	3600	130	133%	81.35	40.756	0.319	4.68	0.37	0.003	0.093	0.354
500	500	3600	130	133%	81.35	40.756	0.1627	4.68	0.37	0.003	0.093	0.182
10	750	3600	130	133%	81.35	40.756	8.1353	4.68	0.37	0.003	0.093	8.952

EDS SP W/kg	EDS SE W-hr/kg	PDS SP W/kg	PDS SE W-hr/kg	diff %	Pwr EDS W	Engy EDS W-hr	mass EDS kg	Pwr PDS W	Engy PDS W-hr	mass PDS kg	PDS time hr	Tot. mass kg
255	750	3600	130	133%	81.35	40.756	0.319	4.68	0.37	0.003	0.093	0.354
500	750	3600	130	133%	81.35	40.756	0.1627	4.68	0.37	0.003	0.093	0.182
10	1000	3600	130	133%	81.35	40.756	8.1353	4.68	0.37	0.003	0.093	8.952
255	1000	3600	130	133%	81.35	40.756	0.319	4.68	0.37	0.003	0.093	0.354
500	1000	3600	130	133%	81.35	40.756	0.1627	4.68	0.37	0.003	0.093	0.182
10	500	6000	130	133%	81.35	40.756	8.1353	4.68	0.37	0.003	0.093	8.952
255	500	6000	130	133%	81.35	40.756	0.319	4.68	0.37	0.003	0.093	0.354
500	500	6000	130	133%	81.35	40.756	0.1627	4.68	0.37	0.003	0.093	0.182
10	750	6000	130	133%	81.35	40.756	8.1353	4.68	0.37	0.003	0.093	8.952
255	750	6000	130	133%	81.35	40.756	0.319	4.68	0.37	0.003	0.093	0.354
500	750	6000	130	133%	81.35	40.756	0.1627	4.68	0.37	0.003	0.093	0.182
10	1000	6000	130	133%	81.35	40.756	8.1353	4.68	0.37	0.003	0.093	8.952
255	1000	6000	130	133%	81.35	40.756	0.319	4.68	0.37	0.003	0.093	0.354
500	1000	6000	130	133%	81.35	40.756	0.1627	4.68	0.37	0.003	0.093	0.182
10	500	1200	200	133%	81.35	40.756	8.1353	4.68	0.37	0.004	0.093	8.953
255	500	1200	200	133%	81.35	40.756	0.319	4.68	0.37	0.004	0.093	0.355
500	500	1200	200	133%	81.35	40.756	0.1627	4.68	0.37	0.004	0.093	0.183
10	750	1200	200	133%	81.35	40.756	8.1353	4.68	0.37	0.004	0.093	8.953
255	750	1200	200	133%	81.35	40.756	0.319	4.68	0.37	0.004	0.093	0.355
500	750	1200	200	133%	81.35	40.756	0.1627	4.68	0.37	0.004	0.093	0.183
10	1000	1200	200	133%	81.35	40.756	8.1353	4.68	0.37	0.004	0.093	8.953
255	1000	1200	200	133%	81.35	40.756	0.319	4.68	0.37	0.004	0.093	0.355
500	1000	1200	200	133%	81.35	40.756	0.1627	4.68	0.37	0.004	0.093	0.183
10	500	3600	200	133%	81.35	40.756	8.1353	4.68	0.37	0.002	0.093	8.951
255	500	3600	200	133%	81.35	40.756	0.319	4.68	0.37	0.002	0.093	0.353
500	500	3600	200	133%	81.35	40.756	0.1627	4.68	0.37	0.002	0.093	0.181
10	750	3600	200	133%	81.35	40.756	8.1353	4.68	0.37	0.002	0.093	8.951
255	750	3600	200	133%	81.35	40.756	0.319	4.68	0.37	0.002	0.093	0.353
500	750	3600	200	133%	81.35	40.756	0.1627	4.68	0.37	0.002	0.093	0.181
10	1000	3600	200	133%	81.35	40.756	8.1353	4.68	0.37	0.002	0.093	8.951
255	1000	3600	200	133%	81.35	40.756	0.319	4.68	0.37	0.002	0.093	0.353
500	1000	3600	200	133%	81.35	40.756	0.1627	4.68	0.37	0.002	0.093	0.181
10	500	6000	200	133%	81.35	40.756	8.1353	4.68	0.37	0.002	0.093	8.951
255	500	6000	200	133%	81.35	40.756	0.319	4.68	0.37	0.002	0.093	0.353
500	500	6000	200	133%	81.35	40.756	0.1627	4.68	0.37	0.002	0.093	0.181
10	750	6000	200	133%	81.35	40.756	8.1353	4.68	0.37	0.002	0.093	8.951
255	750	6000	200	133%	81.35	40.756	0.319	4.68	0.37	0.002	0.093	0.353
500	750	6000	200	133%	81.35	40.756	0.1627	4.68	0.37	0.002	0.093	0.181
10	1000	6000	200	133%	81.35	40.756	8.1353	4.68	0.37	0.002	0.093	8.951
255	1000	6000	200	133%	81.35	40.756	0.319	4.68	0.37	0.002	0.093	0.353
500	1000	6000	200	133%	81.35	40.756	0.1627	4.68	0.37	0.002	0.093	0.181

Appendix D: Relevant Code

D.1. AVL Related Files

a. avl_genmav_batch.bat

File title: avl_genmav_batch.bat
Parent program: AVL
Description: This single-line ".bat" file is the command line execution line. It tells the PC to run AVL with the ".avl" file with the interface keystrokes listed in the ".batch" file. Note that "%" is not a comment designator.
Code as follows:

```
% avl genmav_2pt2.avl genmav_2pt2.run < constraints_genmav.batch
```

b. AVL_MAV_genmav.fileWrapper

File title: AVL_MAV_genmav.fileWrapper
Parent program: Model Center
Description: This is the file used to interface AVL with Model Center. It tells Model Center exactly which variables exist, defines the variables within Model center, points to what they are, and where to look for changed data. This file identifies the input and output file(s). A "fileWrapper" is essentially a plug-in for an executable batch mode program. It requires the Analysis Server.
Code as follows:

```
#
# Athena Vortex Lattice filewrapper
#
# @author: Paul Hrad
# @version: 22 Oct 2009-
# @description: GenMAV AVL (Normal Dihedral - v2.2)
#

RunCommands
{
# Put ModelCenter values in the input file
# Geometry file
 generate inputFile1
#Constraints file
 generate inputFile2
#Run file --trying to run w/o run file since no values in run are user defined.
# generate inputFile3

 # Run the code
# This file transmits instruction to command line
 run "avl_genmav_batch.bat"

# Parse the standard output file
 parse outputfile
}

RowFieldInputFile inputFile1
{
# The template file is not changed. It is a template for the .avl file, which is changed by MS every iteration.
templateFile:			genmav_2pt2.template
initializationFile: genmav_2pt2.initial
fileToGenerate:	genmav_2pt2.avl

setDelimiters " ,"
```

```
setGroup "UserInputs.Geometry_input_file"
variable: Mach              double 6 1      description="keep it zero for M<0.2"
variable: iYsym             double 7 1      description="Symm about y=0 and/or Z=0"
variable: iZsym             double 7 2      description="Symm about y=0 and/or Z=0"
variable: Zsym              double 7 3      description="Symm about y=0 and/or Z=0"
variable: S_ref_sqin        double 8 1      description="reference ares-> all coeffs"
variable: C_ref_in          double 8 2      description="-> Cm"
variable: b_ref_in          double 8 3      description="span-> Cl,Cn"
variable: X_origin_in       double 9 1      description="Global x Point of Origin"
variable: Y_origin_in       double 9 2      description="Global y Point of Origin"
variable: Z_origin_in       double 9 3      description="Global z Point of Origin"
variable: Cdp               double 10 1     description="Optional default profile drag added to geometry"


setGroup "UserInputs.Fuselage"
variable: F_nodes           double 16 1     description="Source-Line node"
variable: F_space           double 16 2     description="Spacing, -3<range<3: 1=cosine"
variable: F_Trans_x         double 19 1     description="Translates fuselage x coordinates"
variable: F_Trans_y         double 19 2     description="Translates fuselage y coordinates"
variable: F_Trans_z         double 19 3     description="Translates fuselage z coordinates"

setGroup "UserInputs.Wing"
variable: Nchord_w          double 29 1     description="# chordwise segments, wing"
variable: Cspace_w          double 29 2     description="Spacing, -3<range<3: 1=cosine"
variable: Nspan_w           double 29 3     description="# spanwise segments, wing"
variable: Sspace_w          double 29 4     description="Spacing, -3<range<3: -2= -sine"
variable: Angle_w           double 31 1     description="Offset added to incidence angles for all sections, in deg"
variable: scale_x_w         double 33 1     description="Scale factor of x coordinates for wing"
variable: scale_y_w         double 33 2     description="Scale factor of y coordinates for wing"
variable: scale_z_w         double 33 3     description="Scale factor of z coordinates for wing"
variable: W_Trans_x         double 35 1     description="Translates wing x coordinates"
variable: W_Trans_y         double 35 2     description="Translates wing y coordinates"
variable: W_Trans_z         double 35 3     description="Translates wing z coordinates"
variable: W_yduplicate      double 37 1     description="y position of duplication"

setGroup "UserInputs.Wing.Section_1"
variable: Xle1              double 43 1     description="Leading Edge Location,x"
variable: Yle1              double 43 2     description="Leading Edge Location,y"
variable: Zle1              double 43 3     description="Leading Edge Location,z"
variable: Chord1            double 43 4     description="Chord length"
variable: Ainc1     double 43 5     description="Incidence angle about spanwise axis projected on y-z plane"

setGroup "UserInputs.Wing.Section_2"
variable: Xle12             double 49 1     description="Leading Edge Location,x"
variable: Yle12             double 49 2     description="Leading Edge Location,y"
variable: Zle12             double 49 3     description="Leading Edge Location,z"
variable: Chord2            double 49 4     description="Chord length"
variable: Ainc2     double 49 5     description="Incidence angle about spanwise axis projected on y-z plane"

setGroup "UserInputs.Wing.Section_3"
variable: Xle3              double 54 1     description="Leading Edge Location,x"
variable: Yle3              double 54 2     description="Leading Edge Location,y"
variable: Zle3              double 54 3     description="Leading Edge Location,z"
variable: Chord3            double 54 4     description="Chord length"
variable: Ainc3     double 54 5     description="Incidence angle about spanwise axis projected on y-z plane"


setGroup "UserInputs.Wing.Section_4"
variable: Xle4              double 59 1     description="Leading Edge Location,x"
variable: Yle4              double 59 2     description="Leading Edge Location,y"
```

```
variable: Zle4          double 59 3     description="Leading Edge Location,z"
variable: Chord4        double 59 4     description="Chord length"
variable: Ainc4   double 59 5     description="Incidence angle about spanwise axis projected on y-z plane"

setGroup "UserInputs.Wing.Section_5"
variable: Xle5          double 64 1     description="Leading Edge Location,x"
variable: Yle5          double 64 2     description="Leading Edge Location,y"
variable: Zle5          double 64 3     description="Leading Edge Location,z"
variable: Chord5        double 64 4     description="Chord length"
variable: Ainc5   double 64 5     description="Incidence angle about spanwise axis projected on y-z plane"

setGroup "UserInputs.Wing.Section_6"
variable: Xle6          double 69 1     description="Leading Edge Location,x"
variable: Yle6          double 69 2     description="Leading Edge Location,y"
variable: Zle6          double 69 3     description="Leading Edge Location,z"
variable: Chord6        double 69 4     description="Chord length"
variable: Ainc6   double 69 5     description="Incidence angle about spanwise axis projected on y-z plane"

setGroup "UserInputs.Wing.Section_7"
variable: Xle7          double 74 1     description="Leading Edge Location,x"
variable: Yle7          double 74 2     description="Leading Edge Location,y"
variable: Zle7          double 74 3     description="Leading Edge Location,z"
variable: Chord7        double 74 4     description="Chord length"
variable: Ainc7   double 74 5     description="Incidence angle about spanwise axis projected on y-z plane"

setGroup "UserInputs.Wing.Section_8"
variable: Xle8          double 79 1     description="Leading Edge Location,x"
variable: Yle8          double 79 2     description="Leading Edge Location,y"
variable: Zle8          double 79 3     description="Leading Edge Location,z"
variable: Chord8        double 79 4     description="Chord length"
variable: Ainc8   double 79 5     description="Incidence angle about spanwise axis projected on y-z plane"

setGroup "UserInputs.Wing.Section_9"
variable: Xle9          double 84 1     description="Leading Edge Location,x"
variable: Yle9          double 84 2     description="Leading Edge Location,y"
variable: Zle9          double 84 3     description="Leading Edge Location,z"
variable: Chord9        double 84 4     description="Chord length"
variable: Ainc9   double 84 5     description="Incidence angle about spanwise axis projected on y-z plane"

setGroup "UserInputs.Wing.Section_10"
variable: Xle10         double 89 1     description="Leading Edge Location,x"
variable: Yle10         double 89 2     description="Leading Edge Location,y"
variable: Zle10         double 89 3     description="Leading Edge Location,z"
variable: Chord10       double 89 4     description="Chord length"
variable: Ainc10  double 89 5     description="Incidence angle about spanwise axis projected on y-z plane"

setGroup "UserInputs.Wing.Section_11"
variable: Xle11         double 94 1     description="Leading Edge Location,x"
variable: Yle11         double 94 2     description="Leading Edge Location,y"
variable: Zle11         double 94 3     description="Leading Edge Location,z"
variable: Chord11       double 94 4     description="Chord length"
variable: Ainc11  double 94 5     description="Incidence angle about spanwise axis projected on y-z plane"

setGroup "UserInputs.Wing.Section_12"
variable: Xle12         double 99 1     description="Leading Edge Location,x"
variable: Yle12         double 99 2     description="Leading Edge Location,y"
variable: Zle12         double 99 3     description="Leading Edge Location,z"
variable: Chord12       double 99 4     description="Chord length"
variable: Ainc12  double 99 5     description="Incidence angle about spanwise axis projected on y-z plane"
```

```
setGroup "UserInputs.Wing.Section_13"
variable: Xle13            double 104 1      description="Leading Edge Location,x"
variable: Yle13            double 104 2      description="Leading Edge Location,y"
variable: Zle13            double 104 3      description="Leading Edge Location,z"
variable: Chord13          double 104 4      description="Chord length"
variable: Ainc13    double 104 5      description="Incidence angle about spanwise axis projected on y-z plane"

setGroup "UserInputs.Wing.Section_14"
variable: Xle14            double 109 1      description="Leading Edge Location,x"
variable: Yle14            double 109 2      description="Leading Edge Location,y"
variable: Zle14            double 109 3      description="Leading Edge Location,z"
variable: Chord14          double 109 4      description="Chord length"
variable: Ainc14    double 109 5      description="Incidence angle about spanwise axis projected on y-z plane"

setGroup "UserInputs.Horizontal_Tail"
variable: Nchord_HT        double 119 1      description="# chordwise segments,HT"
variable: Cspace_HT        double 119 2      description="Spacing, -3<range<3: 1=cosine"
variable: Nspan_HT         double 119 3      description="# spanwise segments,HT"
variable: Sspace_HT        double 119 4      description="Spacing, -3<range<3: -2= -sine"
variable: Angle_HT         double 121 1      description="Offset added to incidence angles for all sections, in deg"
variable: HT_Trans_x       double 123 1      description="Translates HT x coordinates"
variable: HT_Trans_y       double 123 2      description="Translates HT y coordinates"
variable: HT_Trans_z       double 123 3      description="Translates HT z coordinates"
#variable: Y_duplicate     double 125 1      description="Geometric surface reflection about x-z axis"

setGroup "UserInputs.Horizontal_Tail.Left_tip"
variable: Xle1             double 127 1      description="Leading Edge Location,x"
variable: Yle1             double 127 2      description="Leading Edge Location,y"
variable: Zle1             double 127 3      description="Leading Edge Location,z"
variable: Chord1           double 127 4      description="Chord Length"
variable: Ainc1     double 127 5      description="Incidence angle about spanwise axis projected on y-z plane"
variable: L_Elevon_gain1   double 130 2      description="Control deflection gain, units"
variable: L_Elevon_Xhinge1 double 130 3      description="Location of elevator as chord fraction"
#variable: SgnDup          double 130 7  description="Control surface duplication (+1 for elevator)"

setGroup "UserInputs.Horizontal_Tail.Center"
variable: Xle2             double 133 1      description="Leading Edge Location,x"
variable: Yle2             double 133 2      description="Leading Edge Location,y"
variable: Zle2             double 133 3      description="Leading Edge Location,z"
variable: Chord2           double 133 4      description="Chord Length"
variable: Ainc2     double 133 5      description="Incidence angle about spanwise axis projected on y-z plane"
variable: L_Elevon_gain2   double 136 2      description="Control deflection gain, units"
variable: L_Elevon_Xhinge2 double 136 3      description="Location of elevator as chord fraction"
variable: R_Elevon_gain2   double 139 2      description="Control deflection gain, units"
variable: R_Elevon_Xhinge2 double 139 3      description="Location of elevator as chord fraction"

setGroup "UserInputs.Horizontal_Tail.Right_tip"
variable: Xle3             double 142 1      description="Leading Edge Location,x"
variable: Yle3             double 142 2      description="Leading Edge Location,y"
variable: Zle3             double 142 3      description="Leading Edge Location,z"
variable: Chord3           double 142 4      description="Chord Length"
variable: Ainc3     double 142 5      description="Incidence angle about spanwise axis projected on y-z plane"
variable: R_Elevon_gain3   double 145 2      description="Control deflection gain, units"
variable: R_Elevon_Xhinge3 double 145 3      description="Location of elevator as chord fraction"


setGroup "UserInputs.Vertical_Tail"
variable: Nchord_VT        double 149 1      description="# chordwise segments,VT"
variable: Cspace_VT        double 149 2      description="Spacing, -3<range<3: 1=cosine"
variable: Angle_VT         double 151 1      description="Offset added to incidence angles for all sections, in deg"
```

```
variable: VT_Trans_x          double 153 1      description="Translates VT x coordinates"
variable: VT_Trans_y          double 153 2      description="Translates VT y coordinates"
variable: VT_Trans_z          double 153 3      description="Translates VT z coordinates"

setGroup "UserInputs.Vertical_Tail.VT_Root"
variable: Xle1                double 157 1      description="Leading Edge Location,x"
variable: Yle1                double 157 2      description="Leading Edge Location,y"
variable: Zle1                double 157 3      description="Leading Edge Location,z"
variable: Chord1              double 157 4      description="Chord Length"
variable: Ainc1     double 157 5      description="Incidence angle about spanwise axis projected on y-z plane"
variable: Nspan1              double 157 6      description="# spanwise segments,VT"
variable: Sspace1             double 157 7      description="Spacing, -3<range<3: -2= -sine"

setGroup "UserInputs.Vertical_Tail.VT_Tip"
variable: Xle2                double 161 1      description="Leading Edge Location,x"
variable: Yle2                double 161 2      description="Leading Edge Location,y"
variable: Zle2                double 161 3      description="Leading Edge Location,z"
variable: Chord2              double 161 4      description="Chord Length"
variable: Ainc2     double 161 5      description="Incidence angle about spanwise axis projected on y-z plane"
variable: Nspan2              double 161 6      description="# spanwise segments,VT"
variable: Sspace2             double 161 7      description="Spacing, -3<range<3: -2= -sine"
}

RowFieldInputFile inputFile2
{
# The template file is not changed. It is a template for the batch file, which is changed by MS every iteration.
templateFile:                 constraints_genmav.template
fileToGenerate:               constraints_genmav.batch

setDelimiters "= ,"
setGroup UserInputs.Run_Constraints
variable: AirDensity          double 4 2 description="Air Density" units="kg/m^2)"
variable: Velocity            double 5 2 description="Velocity" units="m/s"
variable: Mass                double 3 2 description="Unit Mass" Units="Kg"
variable: PitchMoment         double 7 3 description="Set Pitch Moment"
variable: RollMoment          double 8 3 description="Set Roll Moment"
# The output file is named within this input file.
}

RowFieldOutputFile outputFile
{
# This routine parses the program output file.
# Other variables can be extracted as desired.
 fileToParse: results_genmav.txt

 setDelimiters "= ,"

setGroup Results.Ref_values
markAsBeginning "Sref"
variable: Sref_sqin           double 1 2
variable: Cref_in             double 1 4
variable: bref_in             double 1 6
variable: Xref_CG_in          double 2 2
variable: Yref_CG_in          double 2 4
variable: Zref_CG_in          double 2 6

setGroup Results.Flight_Conditions
variable: Alpha                double 8 2
variable: Beta                 double 9 2
```

```
markAsBeginning "CLtot"
 setGroup Results.Run_Case_AERO_coeff

 variable: CL_tot               double 1 2          description="Total Lift Coef"
 variable: CD_tot               double 2 2          description="Total Drag Coef"
 variable: CD_ind               double 3 4          description="Induced Drag Coef"
 #variable: CL_ff               double 4 2          description="Treftz Lift Ceof"
 #variable: CD_ff               double 4 4          description="Treftz Drag Ceof"
 variable: e                    double 5 4          description="Span Efficiency Factor"

setGroup Results.Conrol_surface_deflections
 variable: L_Elevon             double 7 2          description="Left Elevon deflection"
 variable: R_Elevon             double 8 2          description="Right Elevon deflection"

              setGroup Results.Stability_axis_derivatives
 markAsBeginning "CLa ="

 variable: CL_a                 double 1 5          description="z' force CL"
 variable: Cy_a                 double 2 5          description="y force Cy"
 variable: Cl_a                 double 3 5          description="x' moment Cl"
 variable: Cm_a                 double 4 5          description="y moment Cm"
 variable: cn_a                 double 5 5          description="z' moment Cn"
 variable: CL_b                 double 1 7          description=" "
 variable: Cy_b                 double 2 7          description=" "
 variable: Cl_b                 double 3 7          description=" "
 variable: Cm_b                 double 4 7          description=" "
 variable: cn_b                 double 5 7          description=" "
 variable: Xnp                  double 27 4         description="neutral point"
}
# END AVL_MAV_genmav.fileWrapper
```

c. constraints_genmav.batch

```
Filename: constraints_genmav.batch
Parent program: AVL
Description: This file lists the keystroke entry to do the designated sequence of operations for a single iteration of AVL.
It currently goes to the steady-level flight menu, inputs a mass, density, and velocity.  It tells AVL to deflect the
elevons to provide zero pitch moment and roll moment. Then it executes the integration, calculates stability derivatives,
saves the results, and exits AVL.
Code as follows:
OPER
C1
M 1.018
D 1.225
V 25.0

D1 PM 0.0
D2 RM 0.0
X
ST
results_genmav.txt
Y

QUIT
```

d. constraints_genmav.template

```
Filename: constraints_genmav.template
```

Parent program: Model Center
Description: Same as for "constraints_genmav.batch", except this is the file that does not change. Model Center writes to the ".batch" file for execution of Model Center variable inputs, but keeps a ".template" file as the original Codes as follows: See code for "constraints_genmav.batch"

e. genmav_2pt2.avl

Filename: genmav_2pt2.avl
Parent program: AVL
Description: This is the geometry file used to define the GenMAV for AVL.
Code as follows:

```
GenMAV ! (Normal Dihedral - v2.2)
# Units in inches, sq-in, deg
#==============================================
# HEADER DATA
#==============================================
0.0                            Mach  !Default Mach for Prandtl-Glauert compressibility correction
0.0      0.0      0.0          iYsym  iZsym  Zsym  !Symm about y=0 and/or z=Zsym
111.68  4.7757  24.0           Sref   Cref   Bref  !Sref -> all coeffs | Cref -> Cm | Bref -> Cl,Cn
0.0       0.0      0.0         Xref   Yref   Zref  !Point of Origin
0.102                          CDp !Optional - default profile drag coeff added to geometry
#==============================================
# FUSELAGE DATA
#==============================================
BODY
Fuselage  !body name string
   37.0  1.0  !15 source-line nodes with cosine spacing (1.0)

TRANSLATE
   0.0  0.0  0.0

BFILE
   Fuselage_v2pt2.dat

#==============================================
# WING DATA
#==============================================
SURFACE
Right Wing  !surface name string
   7.0  1.0  25.0  -2.0  !Nchord Cspace Nspan Sspace
ANGLE
   9.0  !Offset added to incidence angles for all sections, in deg
SCALE
   1.0  1.0  1.0
TRANSLATE
   5.75  0.0  1.578
YDUPLICATE
   0.0

#------------------------------------------------------
#   Xle       Yle       Zle       Chord      Ainc
#------------------------------------------------------
SECTION
   0.0       0.0       0.0         5.0       0.0
AFILE
   GenMAVA0.dat
#--------------------------
SECTION
```

```
    0.0        1.5        0.02498    5.0        0.0
AFILE
    GenMAVA0.dat
#---------------------------
SECTION
    0.007623    7.11    0.25615    4.9619     0.0
AFILE
    GenMAVA1.dat
#---------------------------
SECTION
    0.024782    8.23    0.41519    4.8761     0.0
AFILE
    GenMAVA2.dat
#---------------------------
SECTION
    0.047308    8.91    0.54277    4.7635     0.0
AFILE
    GenMAVA3.dat
#---------------------------
SECTION
    0.087735    9.6       0.69949    4.5613     0.0
AFILE
    GenMAVA4.dat
#---------------------------
SECTION
    0.147058    10.2      0.86023    4.2647     0.0
AFILE
    GenMAVA5.dat
#---------------------------
SECTION
    0.249591    10.82     1.05224    3.752      0.0
AFILE
    GenMAVA6.dat
#---------------------------
SECTION
    0.408039    11.37     1.24622    2.9598     0.0
AFILE
    GenMAVA7.dat
#---------------------------
SECTION
    0.512614    11.6      1.33428    2.4369     0.0
AFILE
    GenMAVA8.dat
#---------------------------
SECTION
    0.671564    11.83     1.42659    1.6422     0.0
AFILE
    GenMAVA9.dat
#---------------------------
SECTION
    0.745531    11.9      1.45554    1.2723     0.0
AFILE
    GenMAVA10.dat
#---------------------------
SECTION
    0.818751    11.95     1.47646    0.9062     0.0
AFILE
    GenMAVA11.dat
#---------------------------
SECTION
```

```
    1.0    12.0          1.4976    0.0      0.0
AFILE
    GenMAVA11.dat


#==============================================
# TAIL DATA
#==============================================
SURFACE
Horizontal Tail
    9.0  -2.0   20.0   1.0 !Nchord Cspace Nspan Sspace
ANGLE
    1.5  !Offset added to incidence angles for all sections, in deg
TRANSLATE
    14.75   0.0  0.0
#---------------------------
SECTION  !Horizontal tail fin tip
#    Xle     Yle     Zle    Chord   Ainc
    1.5     -6.0      0.0   2.375   0.0
CONTROL
#    Cname     Cgain   Xhinge  HingeVec
    L_Elevon  1.0     0.5     0.0  0.0  0.0
#---------------------------
SECTION  !Horizontal tail fin root
    0.0   0.0   0.0   0.0   3.875  0.0
CONTROL
#    Cname     Cgain   Xhinge   HingeVec
    L_Elevon  1.0     0.5      0.0  0.0  0.0
CONTROL
#    Cname     Cgain   Xhinge   HingeVec
    R_Elevon  1.0     0.5      0.0  0.0  0.0
#---------------------------
SECTION  !Horizontal tail fin tip
    1.5      6.0       0.0   2.375   0.0
CONTROL
#    Cname     Cgain   Xhinge  HingeVec
    R_Elevon  1.0     0.5     0.0  0.0  0.0
#==============================================
SURFACE
Vertical Tail Fin
    9.0  -2.0  !Nchord Cspace Nspan Sspace
ANGLE
    0.0  !Offset added to incidence angles for all sections, in deg
TRANSLATE
    14.75   0.0  0.0
#---------------------------
SECTION  !Vertical tail fin root
#    Xle     Yle     Zle    Chord   Ainc    Nspan   Sspace
    0.0   0.0   0.0   3.278   0.0     11.0     -2.0

#---------------------------
SECTION  !Vertical tail fin tip
    1.593   0.0   4.625   2.25   0.0     1.0     1.0
#==============================================
# END genmav_2pt2.avl
```

f. Fuselage_v2pt2.dat

```
Filename: Fuselage_v2pt2.dat
```

Parent program: AVL
Description: This is the fuselage geometry. Note that a single column was split into two for page spacing reasons. It should be a single column.
Code as follows:

Fuselage_v2pt2.dat
BATCAM-Like Fuselage

16.5 0.797885	0 -1.055502		
15 0.797885	1 -1.286937		
14 0.797885	2 -1.44186		
13 0.909318	3 -1.596548		
12 1.064884	4 -1.686682		
11 1.180087	5 -1.724589		
10 1.371137	6 -1.691099		
9 1.512565	7 -1.639581		
8 1.586391	8 -1.586391		
7 1.639581	9 -1.512565		
6 1.691099	10 -1.371137		
5 1.724589	11 -1.180087		
4 1.686682	12 -1.064884		
3 1.596548	13 -0.909318		
2 1.44186	14 -0.797885		
1 1.286937	15 -0.797885		
0 1.055502	16.5 -0.797885		

g. GenMAVA0.dat through GenMAVA11.dat

Filename: GenMAVA0.dat through GenMAVA11.dat
Parent program: AVL
Description: Airfoil data from root to tip. Each file is a separate airfoil. Each file is two columns. Four files are presented across the page.
Code as follows:

GenMAVA0.dat		GenMAVA1.dat		GenMAVA2.dat		GenMAVA3.dat	
MAV - v2.0 - Root Airfoil		MAV - v2.0 - Airfoil Cross Section - Chord 4.96188		MAV - v2.0 - Airfoil Cross Section - Chord 4.96188		MAV - v2.0 - Airfoil Cross Section - Chord 4.76346	
1	0.00363	1	-0.00236	1	-0.00236	1	-0.00745
0.98248	0.00465	0.98849	-0.0017	0.98849	-0.0017	0.98432	-0.0068
0.96489	0.00549	0.97076	-0.00085	0.97076	-0.00085	0.96569	-0.00606
0.94721	0.00622	0.95295	-0.00012	0.95295	-0.00012	0.947	-0.0053
0.92947	0.00692	0.93507	0.00059	0.93507	0.00059	0.92826	-0.00447
0.91166	0.00765	0.91713	0.00132	0.91713	0.00132	0.90947	-0.00353
0.89381	0.00844	0.89914	0.00212	0.89914	0.00212	0.89065	-0.00246
0.87591	0.00933	0.8811	0.00302	0.8811	0.00302	0.8718	-0.00124
0.85797	0.01035	0.86303	0.00405	0.86303	0.00405	0.85294	0.00012
0.84002	0.01151	0.84493	0.00521	0.84493	0.00521	0.83409	0.00164

0.82205	0.01281	0.82683	0.00653	0.82683	0.00653	0.81527	0.00331
0.80409	0.01426	0.80873	0.00799	0.80873	0.00799	0.79649	0.00511
0.78616	0.01585	0.79066	0.00959	0.79066	0.00959	0.77777	0.00703
0.76827	0.01756	0.77264	0.01131	0.77264	0.01131	0.75916	0.00904
0.75044	0.01939	0.75466	0.01315	0.75466	0.01315	0.74056	0.01114
0.73271	0.02131	0.7368	0.01509	0.7368	0.01509	0.72199	0.01331
0.71498	0.02331	0.71894	0.0171	0.71894	0.0171	0.70352	0.01552
0.69729	0.02537	0.70111	0.01918	0.70111	0.01918	0.68505	0.01777
0.6797	0.02748	0.68338	0.02131	0.68338	0.02131	0.66659	0.02003
0.6621	0.02962	0.66565	0.02346	0.66565	0.02346	0.64815	0.02229
0.64452	0.03177	0.64794	0.02564	0.64794	0.02564	0.62971	0.02455
0.62695	0.03393	0.63023	0.02781	0.63023	0.02781	0.61122	0.02678
0.60938	0.03608	0.61252	0.02997	0.61252	0.02997	0.59273	0.02898
0.59177	0.03821	0.59478	0.03212	0.59478	0.03212	0.57423	0.03114
0.57415	0.0403	0.57703	0.03423	0.57703	0.03423	0.5557	0.03325
0.55653	0.04236	0.55927	0.0363	0.55927	0.0363	0.53718	0.0353
0.53888	0.04437	0.54148	0.03833	0.54148	0.03833	0.51863	0.0373
0.52122	0.04633	0.52369	0.0403	0.52369	0.0403	0.50007	0.03923
0.50356	0.04823	0.50589	0.04221	0.50589	0.04221	0.48151	0.0411
0.48588	0.05007	0.48807	0.04407	0.48807	0.04407	0.46292	0.04291
0.4682	0.05185	0.47026	0.04586	0.47026	0.04586	0.44432	0.04466
0.45048	0.05358	0.45241	0.0476	0.45241	0.0476	0.42571	0.04635
0.43276	0.05524	0.43455	0.04928	0.43455	0.04928	0.40705	0.04798
0.41503	0.05685	0.41668	0.0509	0.41668	0.0509	0.38839	0.04955
0.39725	0.0584	0.39877	0.05247	0.39877	0.05247	0.3697	0.05105
0.37948	0.0599	0.38086	0.05397	0.38086	0.05397	0.35097	0.05248
0.36167	0.06133	0.36291	0.05542	0.36291	0.05542	0.33225	0.05383
0.34383	0.06269	0.34494	0.05679	0.34494	0.05679	0.31347	0.05508
0.32599	0.06398	0.32696	0.05808	0.32696	0.05808	0.29467	0.05621
0.3081	0.06517	0.30893	0.05929	0.30893	0.05929	0.27588	0.0572
0.2902	0.06625	0.29089	0.06037	0.29089	0.06037	0.25703	0.05799
0.27229	0.06718	0.27284	0.06132	0.27284	0.06132	0.23818	0.05856
0.25433	0.06794	0.25475	0.06208	0.25475	0.06208	0.21933	0.05883
0.23637	0.06848	0.23665	0.06262	0.23665	0.06262	0.20048	0.05873
0.21842	0.06874	0.21856	0.06288	0.21856	0.06288	0.18163	0.05818
0.20046	0.06865	0.20046	0.06279	0.20046	0.06279	0.16286	0.05709
0.1825	0.06812	0.18236	0.06226	0.18236	0.06226	0.14412	0.05532
0.16462	0.06708	0.16434	0.06121	0.16434	0.06121	0.1254	0.05275
0.14677	0.0654	0.14636	0.05952	0.14636	0.05952	0.10703	0.0493
0.12893	0.06295	0.12838	0.05705	0.12838	0.05705	0.08867	0.04475
0.11143	0.05966	0.11075	0.05374	0.11075	0.05374	0.07065	0.03902
0.09394	0.05532	0.09312	0.04937	0.09312	0.04937	0.05324	0.03211
0.07677	0.04986	0.07582	0.04386	0.07582	0.04386	0.03584	0.02363
0.06019	0.04328	0.05911	0.03723	0.05911	0.03723	0.02006	0.0144
0.04361	0.0352	0.04241	0.02909	0.04241	0.02909	0.00506	0.00406
0.02857	0.02641	0.02726	0.02023	0.02726	0.02023	0	0

0.01429	0.01656	0.01286	0.0103	0.01286	0.0103	0	-0.0042
0	0.00511	0	0	0	0	0.00506	-0.00014
0	0.00111	0	-0.00403	0	-0.00403	0.02006	0.0102
0.01429	0.01256	0.01286	0.00627	0.01286	0.00627	0.03584	0.01943
0.02857	0.02241	0.02726	0.0162	0.02726	0.0162	0.05324	0.02791
0.04361	0.0312	0.04241	0.02506	0.04241	0.02506	0.07065	0.03482
0.06019	0.03928	0.05911	0.0332	0.05911	0.0332	0.08867	0.04055
0.07677	0.04586	0.07582	0.03983	0.07582	0.03983	0.10703	0.0451
0.09394	0.05132	0.09312	0.04533	0.09312	0.04533	0.1254	0.04855
0.11143	0.05566	0.11075	0.04971	0.11075	0.04971	0.14412	0.05112
0.12893	0.05895	0.12838	0.05302	0.12838	0.05302	0.16286	0.05289
0.14677	0.0614	0.14636	0.05549	0.14636	0.05549	0.18163	0.05398
0.16462	0.06308	0.16434	0.05718	0.16434	0.05718	0.20048	0.05453
0.1825	0.06412	0.18236	0.05823	0.18236	0.05823	0.21933	0.05463
0.20046	0.06465	0.20046	0.05876	0.20046	0.05876	0.23818	0.05436
0.21842	0.06474	0.21856	0.05885	0.21856	0.05885	0.25703	0.0538
0.23637	0.06448	0.23665	0.05859	0.23665	0.05859	0.27588	0.053
0.25433	0.06394	0.25475	0.05805	0.25475	0.05805	0.29467	0.05201
0.27229	0.06318	0.27284	0.05729	0.27284	0.05729	0.31347	0.05088
0.2902	0.06225	0.29089	0.05634	0.29089	0.05634	0.33225	0.04963
0.3081	0.06117	0.30893	0.05526	0.30893	0.05526	0.35097	0.04828
0.32599	0.05998	0.32696	0.05405	0.32696	0.05405	0.3697	0.04685
0.34383	0.05869	0.34494	0.05276	0.34494	0.05276	0.38839	0.04535
0.36167	0.05733	0.36291	0.05138	0.36291	0.05138	0.40705	0.04378
0.37948	0.0559	0.38086	0.04994	0.38086	0.04994	0.42571	0.04215
0.39725	0.0544	0.39877	0.04844	0.39877	0.04844	0.44432	0.04046
0.41503	0.05285	0.41668	0.04687	0.41668	0.04687	0.46292	0.03871
0.43276	0.05124	0.43455	0.04525	0.43455	0.04525	0.48151	0.0369
0.45048	0.04958	0.45241	0.04357	0.45241	0.04357	0.50007	0.03503
0.4682	0.04785	0.47026	0.04183	0.47026	0.04183	0.51863	0.0331
0.48588	0.04607	0.48807	0.04004	0.48807	0.04004	0.53718	0.0311
0.50356	0.04423	0.50589	0.03818	0.50589	0.03818	0.5557	0.02905
0.52122	0.04233	0.52369	0.03627	0.52369	0.03627	0.57423	0.02694
0.53888	0.04037	0.54148	0.0343	0.54148	0.0343	0.59273	0.02478
0.55653	0.03836	0.55927	0.03227	0.55927	0.03227	0.61122	0.02258
0.57415	0.0363	0.57703	0.0302	0.57703	0.0302	0.62971	0.02035
0.59177	0.03421	0.59478	0.02809	0.59478	0.02809	0.64815	0.0181
0.60938	0.03208	0.61252	0.02594	0.61252	0.02594	0.66659	0.01583
0.62695	0.02993	0.63023	0.02378	0.63023	0.02378	0.68505	0.01357
0.64452	0.02777	0.64794	0.0216	0.64794	0.0216	0.70352	0.01132
0.6621	0.02562	0.66565	0.01943	0.66565	0.01943	0.72199	0.00911
0.6797	0.02348	0.68338	0.01728	0.68338	0.01728	0.74056	0.00694
0.69729	0.02137	0.70111	0.01515	0.70111	0.01515	0.75916	0.00484
0.71498	0.01931	0.71894	0.01307	0.71894	0.01307	0.77777	0.00283
0.73271	0.01731	0.7368	0.01106	0.7368	0.01106	0.79649	0.00091
0.75044	0.01539	0.75466	0.00912	0.75466	0.00912	0.81527	-0.00089

0.76827	0.01356	0.77264	0.00728	0.77264	0.00728	0.83409	-0.00255
0.78616	0.01185	0.79066	0.00556	0.79066	0.00556	0.85294	-0.00407
0.80409	0.01026	0.80873	0.00396	0.80873	0.00396	0.8718	-0.00544
0.82205	0.00881	0.82683	0.0025	0.82683	0.0025	0.89065	-0.00666
0.84002	0.00751	0.84493	0.00118	0.84493	0.00118	0.90947	-0.00773
0.85797	0.00635	0.86303	0.00002	0.86303	0.00002	0.92826	-0.00866
0.87591	0.00533	0.8811	-0.00101	0.8811	-0.00101	0.947	-0.0095
0.89381	0.00444	0.89914	-0.00191	0.89914	-0.00191	0.96569	-0.01026
0.91166	0.00365	0.91713	-0.00271	0.91713	-0.00271	0.98432	-0.01099
0.92947	0.00292	0.93507	-0.00344	0.93507	-0.00344		
0.94721	0.00222	0.95295	-0.00415	0.95295	-0.00415		
0.96489	0.00149	0.97076	-0.00489	0.97076	-0.00489		
0.98248	0.00065	0.98849	-0.00573	0.98849	-0.00573		

GenMAVA4.dat		GenMAVA5.dat		GenMAVA6.dat		GenMAVA7.dat	
MAV - v2.0 - Airfoil Cross Section - Chord 4.56133		MAV - v2.0 - Airfoil Cross Section - Chord 4.26471		MAV - v2.0 - Airfoil Cross Section - Chord 3.75205		MAV - v2.0 - Airfoil Cross Section - Chord 2.95981	
1	-0.01304	1	-0.02097	1	-0.03156	1	-0.03916
0.99962	-0.01303	0.99244	-0.02059	0.98112	-0.02989	0.98063	-0.0368
0.98011	-0.01223	0.97142	-0.0194	0.95728	-0.02761	0.95093	-0.03316
0.96053	-0.01136	0.95036	-0.01804	0.93352	-0.02517	0.92125	-0.02951
0.94091	-0.01038	0.9293	-0.01652	0.90989	-0.02262	0.89156	-0.02589
0.92125	-0.00927	0.90825	-0.01482	0.88627	-0.01995	0.86181	-0.02229
0.90157	-0.008	0.88722	-0.01296	0.8627	-0.0172	0.83205	-0.01875
0.88187	-0.00657	0.86625	-0.01095	0.83925	-0.01439	0.80228	-0.01528
0.86219	-0.00498	0.84534	-0.00881	0.8158	-0.01154	0.77246	-0.01188
0.84253	-0.00324	0.82455	-0.00656	0.79237	-0.00867	0.74264	-0.00858
0.82293	-0.00137	0.80377	-0.00421	0.76896	-0.00579	0.7128	-0.00537
0.80337	0.00064	0.78303	-0.00179	0.74554	-0.00293	0.68293	-0.00226
0.78394	0.00274	0.7624	0.00068	0.72207	-0.00009	0.65306	0.00075
0.76451	0.00493	0.74177	0.00319	0.6986	0.0027	0.62314	0.00367
0.74512	0.0072	0.72116	0.00572	0.67511	0.00544	0.5932	0.00648
0.72583	0.0095	0.70056	0.00825	0.65159	0.00812	0.56325	0.0092
0.70654	0.01185	0.67996	0.01076	0.62807	0.01073	0.53322	0.01182
0.68727	0.01421	0.65931	0.01326	0.60452	0.01326	0.50319	0.01435
0.66801	0.01658	0.63866	0.01572	0.58096	0.01571	0.47311	0.01676
0.64875	0.01893	0.618	0.01813	0.5574	0.01809	0.44297	0.01907
0.62944	0.02127	0.5973	0.02048	0.53379	0.02039	0.41284	0.02124
0.61013	0.02356	0.57661	0.02278	0.51018	0.02261	0.38261	0.02325
0.59082	0.02582	0.5559	0.025	0.48655	0.02475	0.35237	0.02508
0.57147	0.02802	0.53517	0.02716	0.46286	0.02682	0.32211	0.02666
0.55212	0.03017	0.51444	0.02925	0.43917	0.02881	0.29178	0.02794
0.53275	0.03225	0.49367	0.03128	0.41544	0.03072	0.26145	0.02885
0.51337	0.03427	0.47289	0.03323	0.39167	0.03253	0.23111	0.02928

0.49399	0.03622	0.4521	0.03512	0.3679	0.03425	0.20077	0.02913
0.47457	0.03811	0.43126	0.03694	0.34406	0.03584	0.17043	0.02825
0.45514	0.03994	0.41042	0.03869	0.3202	0.03727	0.14023	0.02648
0.43571	0.0417	0.38955	0.04036	0.29633	0.03852	0.11007	0.02364
0.41622	0.0434	0.36863	0.04196	0.2724	0.03954	0.07994	0.01951
0.39674	0.04504	0.34771	0.04347	0.24847	0.04025	0.05038	0.01395
0.37722	0.04661	0.32674	0.04487	0.22454	0.04059	0.02083	0.00662
0.35766	0.0481	0.30575	0.04613	0.20061	0.04047	0	0
0.33811	0.04951	0.28475	0.04723	0.17668	0.03977	0	-0.00676
0.3185	0.05082	0.2637	0.04812	0.15285	0.03838	0.02083	-0.00013
0.29887	0.052	0.24265	0.04875	0.12906	0.03614	0.05038	0.0072
0.27924	0.05303	0.22159	0.04905	0.10529	0.03288	0.07994	0.01275
0.25956	0.05386	0.20054	0.04895	0.08197	0.0285	0.11007	0.01688
0.23987	0.05445	0.17948	0.04833	0.05866	0.02272	0.14023	0.01972
0.22019	0.05473	0.15852	0.04711	0.03578	0.01544	0.17043	0.02149
0.2005	0.05463	0.13759	0.04514	0.01368	0.00667	0.20077	0.02237
0.18081	0.05406	0.11667	0.04227	0	0	0.23111	0.02253
0.16121	0.05291	0.09616	0.03841	0	-0.00533	0.26145	0.02209
0.14165	0.05107	0.07565	0.03333	0.01368	0.00134	0.29178	0.02119
0.12209	0.04839	0.05552	0.02692	0.03578	0.01011	0.32211	0.0199
0.10291	0.04478	0.03608	0.01921	0.05866	0.01739	0.35237	0.01832
0.08374	0.04003	0.01664	0.00973	0.08197	0.02317	0.38261	0.0165
0.06491	0.03404	0	0	0.10529	0.02755	0.41284	0.01448
0.04674	0.02683	0	-0.00469	0.12906	0.03081	0.44297	0.01231
0.02857	0.01797	0.01664	0.00505	0.15285	0.03305	0.47311	0.01001
0.01209	0.00833	0.03608	0.01452	0.17668	0.03444	0.50319	0.00759
0	0	0.05552	0.02223	0.20061	0.03514	0.53322	0.00507
0	-0.00438	0.07565	0.02864	0.22454	0.03526	0.56325	0.00245
0.01209	0.00395	0.09616	0.03372	0.24847	0.03492	0.5932	-0.00027
0.02857	0.01359	0.11667	0.03758	0.2724	0.0342	0.62314	-0.00309
0.04674	0.02244	0.13759	0.04045	0.29633	0.03319	0.65306	-0.006
0.06491	0.02966	0.15852	0.04242	0.3202	0.03194	0.68293	-0.00901
0.08374	0.03564	0.17948	0.04364	0.34406	0.03051	0.7128	-0.01213
0.10291	0.0404	0.20054	0.04426	0.3679	0.02892	0.74264	-0.01533
0.12209	0.044	0.22159	0.04436	0.39167	0.0272	0.77246	-0.01864
0.14165	0.04669	0.24265	0.04406	0.41544	0.02539	0.80228	-0.02203
0.16121	0.04853	0.2637	0.04343	0.43917	0.02348	0.83205	-0.02551
0.18081	0.04967	0.28475	0.04254	0.46286	0.02149	0.86181	-0.02905
0.2005	0.05025	0.30575	0.04144	0.48655	0.01942	0.89156	-0.03265
0.22019	0.05035	0.32674	0.04018	0.51018	0.01728	0.92125	-0.03627
0.23987	0.05007	0.34771	0.03878	0.53379	0.01506	0.95093	-0.03992
0.25956	0.04948	0.36863	0.03727	0.5574	0.01276	0.98063	-0.04356
0.27924	0.04865	0.38955	0.03567	0.58096	0.01038		
0.29887	0.04762	0.41042	0.034	0.60452	0.00793		
0.3185	0.04644	0.43126	0.03225	0.62807	0.0054		
0.33811	0.04513	0.4521	0.03043	0.65159	0.00279		

0.35766	0.04372	0.47289	0.02854	0.67511	0.00011		
0.37722	0.04222	0.49367	0.02659	0.6986	-0.00263		
0.39674	0.04066	0.51444	0.02456	0.72207	-0.00542		
0.41622	0.03902	0.53517	0.02247	0.74554	-0.00826		
0.43571	0.03732	0.5559	0.02031	0.76896	-0.01112		
0.45514	0.03555	0.57661	0.01809	0.79237	-0.014		
0.47457	0.03373	0.5973	0.01579	0.8158	-0.01687		
0.49399	0.03184	0.618	0.01344	0.83925	-0.01972		
0.51337	0.02988	0.63866	0.01103	0.8627	-0.02253		
0.53275	0.02786	0.65931	0.00857	0.88627	-0.02528		
0.55212	0.02578	0.67996	0.00607	0.90989	-0.02795		
0.57147	0.02364	0.70056	0.00356	0.93352	-0.0305		
0.59082	0.02143	0.72116	0.00103	0.95728	-0.03294		
0.61013	0.01918	0.74177	-0.0015	0.98112	-0.03522		
0.62944	0.01688	0.7624	-0.00401				
0.64875	0.01455	0.78303	-0.00648				
0.66801	0.0122	0.80377	-0.0089				
0.68727	0.00983	0.82455	-0.01125				
0.70654	0.00747	0.84534	-0.0135				
0.72583	0.00512	0.86625	-0.01564				
0.74512	0.00281	0.88722	-0.01765				
0.76451	0.00055	0.90825	-0.01951				
0.78394	-0.00165	0.9293	-0.02121				
0.80337	-0.00375	0.95036	-0.02273				
0.82293	-0.00575	0.97142	-0.02409				
0.84253	-0.00763	0.99244	-0.02528				
0.86219	-0.00937						
0.88187	-0.01096						
0.90157	-0.01238						
0.92125	-0.01365						
0.94091	-0.01477						
0.96053	-0.01575						
0.98011	-0.01662						
0.99962	-0.01741						

GenMAVA8.dat		GenMAVA9.dat		GenMAVA10.dat		GenMAVA11.dat	
MAV - v2.0 - Airfoil Cross Section - Chord 2.43693		MAV - v2.0 - Airfoil Cross Section - Chord 1.64218		MAV - v2.0 - Airfoil Cross Section - Chord 1.27234		MAV - v2.0 - Airfoil Cross Section - Chord 0.90625	
1	-0.03903	1	-0.03443	1	-0.03052	1	-0.02225
0.96767	-0.03519	0.96265	-0.03079	0.97516	-0.02835	0.99356	-0.02376
0.93151	-0.03097	0.90869	-0.02572	0.9053	-0.02248	0.89513	-0.01667
0.89529	-0.02684	0.85471	-0.02082	0.83533	-0.01686	0.79642	-0.01009
0.85908	-0.02283	0.80058	-0.01609	0.76522	-0.0115	0.69763	-0.00414
0.82283	-0.01893	0.74646	-0.01155	0.69512	-0.00645	0.59882	0.00104

0.78655	-0.01515	0.69225	-0.00719	0.62481	-0.00176	0.49975	0.00523
0.75027	-0.0115	0.63793	-0.00304	0.55445	0.00247	0.40069	0.00819
0.71393	-0.00796	0.58361	0.00087	0.48407	0.00616	0.30161	0.00961
0.67757	-0.00454	0.52914	0.00451	0.4135	0.00914	0.20252	0.0091
0.64119	-0.00123	0.47462	0.00779	0.34294	0.01125	0.10343	0.00622
0.60472	0.00195	0.42009	0.01064	0.27237	0.01226	0.00478	0.00045
0.56824	0.00501	0.36542	0.01296	0.2018	0.0119	0	0
0.53171	0.00795	0.31075	0.01459	0.13122	0.00985	0	-0.02207
0.49511	0.01075	0.25607	0.01537	0.06095	0.00574	0.00478	-0.02162
0.4585	0.01338	0.20139	0.01509	0	0	0.10343	-0.01585
0.4218	0.01583	0.14671	0.0135	0	-0.01572	0.20252	-0.01297
0.38506	0.01804	0.09227	0.01032	0.06095	-0.00998	0.30161	-0.01246
0.34831	0.01997	0.03792	0.0052	0.13122	-0.00587	0.40069	-0.01388
0.31147	0.02153	0	0	0.2018	-0.00382	0.49975	-0.01684
0.27463	0.02263	0	-0.01055	0.27237	-0.00346	0.59882	-0.02103
0.23779	0.02315	0.03792	-0.00698	0.34294	-0.00446	0.69763	-0.0262
0.20094	0.02297	0.09227	-0.00186	0.4135	-0.00657	0.79642	-0.03216
0.16409	0.02189	0.14671	0.00132	0.48407	-0.00956	0.89513	-0.03874
0.1274	0.01975	0.20139	0.00292	0.55445	-0.01324	0.99356	-0.04583
0.09078	0.0163	0.25607	0.0032	0.62481	-0.01748		
0.05418	0.01128	0.31075	0.00241	0.69512	-0.02217		
0.01828	0.00453	0.36542	0.00078	0.76522	-0.02722		
0	0	0.42009	-0.00154	0.83533	-0.03258		
0	-0.00821	0.47462	-0.00439	0.9053	-0.0382		
0.01828	-0.00367	0.52914	-0.00767	0.97516	-0.04407		
0.05418	0.00307	0.58361	-0.01131				
0.09078	0.00809	0.63793	-0.01522				
0.1274	0.01154	0.69225	-0.01937				
0.16409	0.01369	0.74646	-0.02373				
0.20094	0.01476	0.80058	-0.02827				
0.23779	0.01495	0.85471	-0.033				
0.27463	0.01442	0.90869	-0.0379				
0.31147	0.01332	0.96265	-0.04297				
0.34831	0.01176						
0.38506	0.00984						
0.4218	0.00762						
0.4585	0.00517						
0.49511	0.00254						
0.53171	-0.00026						
0.56824	-0.00319						
0.60472	-0.00626						
0.64119	-0.00944						
0.67757	-0.01274						
0.71393	-0.01616						
0.75027	-0.0197						
0.78655	-0.02336						

0.82283	-0.02714							
0.85908	-0.03104							
0.89529	-0.03505							
0.93151	-0.03917							
0.96767	-0.04339							

h. genmav_2pt2.initial

Filename: genmav_2pt2.initial
Parent program: AVL
Description: This is an optional file that has identical format as the ".avl" file used to set initial values.
Code as follows: See genmav_2pt2.avl

i. genmav_2pt2.template

Filename: genmav_2pt2.template
Parent program: Model Center
Description: This is a file used by Model Center that is not changed. The original is maintained in the ".template." The file that is actually changed is genmav_2pt2.avl
Code as follows: See genmav_2pt2.avl

j. genmav_2pt2.mass

Filename: genmav_2pt2.mass
Parent program: AVL
Description: This file provides all information about the airframe mass, coordinates, units, and MOI, etc. A very detailed break down of each subcomponent is not necessary unless user desires Eigenmode analysis.
Code as follows:

```
#
#  GenMAV - version 2.2 - cg 6.4 in
#  0-deg Dihedral (Baseline)
#  Mass & Inertia Breakdown
#
#   x back
#   y right
#   z up
#
#=============================================
# UNIT DATA
# - Scales the mass, xyz, and inertia table data
# - LUNIT also scales all lengths / areas in AVL input file
#=============================================
Lunit = 0.0254 m
Munit = 0.001 kg
Tunit = 1.0 s
#=============================================
# CONSTANTS
# - Gravity and density to be used in trim setup
# - Must be in unit names given above
#=============================================
g   = 9.81
rho = 1.225
#=============================================
# MASS, POSITION, INERTIA DATA
# - x,y,z is location of item's own cg
# - Ixx...Iyz are item's inertias about item's own cg
```

```
# - Must use same point of origin as AVL input file
#===============================================
#  Mass    x        y        z       Ixx       Iyy      Izz
*  1       1        1        1       1         1        1
+  0       0        0        0       0         0        0
   452.00  7.3160  0.000  0.123  141.86  1453.55  1420.45   ! Fuselage
   24.450  7.9870 -5.036  1.505  293.40  37.37    330.77    ! Left Wing
   24.450  7.9870  5.036  1.505  293.40  37.37    330.77    ! Right Wing
   8.3945  17.033 -2.760  0.000  25.180  6.830    32.010    ! Left Horizontal Tail
   8.3945  17.033  2.760  0.000  25.180  6.830    32.010    ! Right Horizontal Tail
   5.1766  16.816  0.000  2.235  9.2300  11.92    2.7000    ! Vertical Tail
   15.200 -0.3750  0.000  0.000  57.480  28.74    28.740    ! Propeller
   44.500  0.5000  0.000  0.000  5.5600  5.620    5.6200    ! Motor
   142.50  3.0000  0.000  0.750  25.110  21.45    7.5700    ! Battery
   91.000  5.2500 -0.563  0.000  81.050  81.05    25.590    ! Left Shock
   91.000  5.1250  0.563  0.000  81.050  81.05    25.590    ! Right Shock
   24.600  7.0000  0.000 -0.500  0.4600  0.980    1.4200    ! Spd Cntrl
   44.600  7.5000  0.000  2.240  1.1600  1.160    1.8600    ! AP / Modem
   16.000  9.8750  0.000  0.750  0.3000  1.030    1.1700    ! GPS
   9.3000  14.625  0.000  0.250  0.1500  0.240    0.1600    ! Servo 1
   9.3000  15.438  0.000  0.188  0.1500  0.240    0.1600    ! Servo 2
   7.1000  17.000  0.000  1.500  0.0700  0.070    0.0700    ! Camera
```

k. genmav_2pt2.run

Filename: genmav_2pt2.run
Parent program: AVL
Description: This optional file is both an input and an output file. Individual flight conditions can be set up and inputted at once through the ".run" file to be called by AVL. Or, the same information can be typed line by line through the AVL user interface. After a flight condition has been executed, the flight condition or setup can be saved for future runs as a ".run" file.
Code as follows:

```
---------------------------------------------
 Run case  1:  Base Run

 alpha       ->  alpha       =   0.00000
 beta        ->  beta        =   0.00000
 pb/2V       ->  pb/2V       =   0.00000
 qc/2V       ->  qc/2V       =   0.00000
 rb/2V       ->  rb/2V       =   0.00000
 L_Elevon    ->  Cm pitchmom =   0.00000
 R_Elevon    ->  Cl roll mom =   0.00000

 alpha     =   0.00000     deg
 beta      =   0.00000     deg
 pb/2V     =  -0.365837E-15
 qc/2V     =   0.00000
 rb/2V     =   0.00000
 CL        =   0.553571
 CDo       =   0.102
 bank      =   0.00000     deg
 elevation =   0.00000     deg
 heading   =   0.00000     deg
 Mach      =   0.400000E-01
 velocity  =   13.4100     m/s
 density   =   1.22500     kg/m^3
 grav.acc. =   9.81000     m/s^2
 turn_rad. =   0.00000     m
```

```
 load_fac. =   1.00000
 X_cg      =   6.40844     m
 Y_cg      =  0.409869E-09 m
 Z_cg      =  0.355575     m
 mass      =   1.01797     kg
 Ixx       =  0.181759E-02 kg-m^2
 Iyy       =  0.754403E-02 kg-m^2
 Izz       =  0.854157E-02 kg-m^2
 Ixy       =  0.413171E-05 kg-m^2
 Iyz       = -0.268490E-11 kg-m^2
 Izx       = -0.137953E-03 kg-m^2
 visc CL_a =   0.00000
 visc CL_u =   0.00000
 visc CM_a =   0.00000
 visc CM_u =   0.00000
```

l. genmav_2pt2run.template

Filename: genmav_2pt2run.template
Parent program: Model Center
Description: This is a file used by Model Center that is not changed. The original is maintained in the ".template." The file that is actually changed is genmav_2pt2.run. Note that I did not use the ".run" file to interface with Model Center, but some cases exist where it would be a useful method.
Code as follows: See genmav_2pt2.avl

m. results_genmav.txt

Filename: results_genmav.txt
Parent program: AVL
Description: AVL can export results to a text based file. The file constraint_genmav.batch named this file. From the ".fileWrapper" file, Model Center knows to retrieve results from this ".txt" file.
Code as follows:

```
---------------------------------------------------------------
 Vortex Lattice Output -- Total Forces

 Configuration: GenMAV
     # Surfaces =   4
     # Strips   =  81
     # Vortices = 629

  Sref =  111.68       Cref =  4.7757       Bref =  24.000
  Xref =  6.4084       Yref = 0.40987E-09   Zref = 0.35557

 Standard axis orientation,  X fwd, Z down

 Run case: Base Run

  Alpha =  -6.49872     pb/2V =   0.00000     p'b/2V =   0.00000
  Beta  =   0.00000     qc/2V =   0.00000
  Mach  =     0.040     rb/2V =   0.00000     r'b/2V =   0.00000

  CXtot =  -0.15160     Cltot =   0.00000     Cl'tot =   0.00000
  CYtot =   0.00000     Cmtot =   0.00000
  CZtot =  -0.34714     Cntot =   0.00000     Cn'tot =   0.00000

  CLtot =   0.36207
  CDtot =   0.11133
  CDvis =   0.10200     CDind =   0.00933
```

```
 CLff =   0.35453     CDff =   0.00965    | Trefftz
 CYff =   0.00000        e =    0.8039    | Plane

  L_Elevon       =  6.40806
  R_Elevon       =  6.40857

 ---------------------------------------------------------------

 Derivatives...
                      alpha             beta
              ----------------    ----------------
 z force     |   CLa =   5.289935    CLb =  -0.000005
 y force     |   CYa =  -0.000001    CYb =  -0.453573
 roll  x mom.|   Cla =  -0.000002    Clb =  -0.139558
 pitch y mom.|   Cma =  -1.588459    Cmb =   0.000009
 yaw   z mom.|   Cna =   0.000000    Cnb =   0.039762

                 roll rate  p      pitch rate  q       yaw rate  r
              ----------------    ----------------    ----------------
 z force     |   CLp =   0.000001    CLq =  10.134153    CLr =   0.000003
 y force     |   CYp =  -0.138981    CYq =  -0.000007    CYr =   0.205479
 roll  x mom.|   Clp =  -0.466106    Clq =  -0.000015    Clr =   0.146314
 pitch y mom.|   Cmp =  -0.000003    Cmq = -12.193177    Cmr =  -0.000006
 yaw   z mom.|   Cnp =  -0.002405    Cnq =  -0.000002    Cnr =  -0.116552

              L_Elevon     d1     R_Elevon     d2
              ----------------    ----------------
 z force     |  CLd1 =   0.009382   CLd2 =   0.009382
 y force     |  CYd1 =  -0.000486   CYd2 =   0.000486
 roll  x mom.|  Cld1 =   0.000641   Cld2 =  -0.000641
 pitch y mom.|  Cmd1 =  -0.017995   Cmd2 =  -0.017996
 yaw   z mom.|  Cnd1 =   0.000209   Cnd2 =  -0.000209
 Trefftz drag| CDffd1 =   0.000097 CDffd2 =   0.000097
 span eff.   |    ed1 =   0.034807    ed2 =   0.034806



 Neutral point  Xnp =   7.842484

 Clb Cnr / Clr Cnb  =   2.795906    (  > 1 if spirally stable )
```

D.2. Velocity, Thrust, Power

a. VeloCalcEnd.m

Filename: VeloCalcEnd.m
Parent program: Model Center and Matlab Plug-in
Description: This Matlab script calculates the endurance velocity, max L/D, and stall velocity. Note that the variable that are commented out on top tell Model Center about the variables.
Code as follows:

```
% variable: C_L_max          double    input description = "Max lift coefficient"
% variable: mass             double    input description = "mass"                  units="kg"
% variable: S_in2            double    input description = "Planform area"         units="in^2"
% variable: den              double    input description = "Density"               units="kg/m^3"
% variable: b_in             double    input description = "reference span"        units="in"
% variable: C_D_0            double    input description = "zero lift drag coef"
% variable: e                double    input description = "AVL output: span efficiency"
% variable: AspectRatio      double    output description =Wing aspect ratio"  matlabName="AR"
```

```
% variable: L_over_D_max     double    output description ="Maximum Lift/Drag"
% variable: v_stall          double    output description ="Stall velocity"                  units="m/s"
% variable: v_end            double    output description ="Endurance or loiter velocity" units="m/s"

% Uncomment the following in order to test the code in Matlab
% close all; clear all; clc;
% S_in2              = 111
% b_in               = 24
% mass               = 1
% den                =1.225
% e                  =0.92
% %velo              =14
% %climb_rate        =0.01
% C_D_0              =0.1
% C_L_max             = 1.2
% g                  = 9.81
% b_m                = b_in*0.0254
% S_m2               = S_in2*0.0254^2
% w                  = mass*g
% AR                 = (b_m^2)/S_m2
% K                  = 1/(pi*e*AR)

g                     = 9.81
b_m                   = b_in*0.0254
S_m2                  = S_in2*0.0254^2
w                     = mass*g
AR                    = (b_m^2)/S_m2
K                     = 1/(pi*e*AR)

v_stall              = sqrt(2*w/(den*S_m2*C_L_max))
L_over_D_max         = 1/(sqrt(4*C_D_0*K))
v_end                = sqrt(2*w*(sqrt(K/(3*C_D_0)))/(den*S_m2))
```

b. VeloCalcPassThrough.m

```
Filename: VeloCalcPassThrough.m
Parent Program: Model Center and Matlab Plug-in
Description: This program passes the user requested velocity and uses AVL outputs to calculate stall speed.
Code as follows:
% variable: C_L_max             double    input     description = "Max lift coefficient"
% variable: mass                double    input     description = "mass"                     units="kg"
% variable: S_in2               double    input     description = "Planform area"            units="in^2"
% variable: den                 double    input     description = "Density"                  units="kg/m^3"
% variable: b_in                double    input     description = "reference span"           units="in"
% variable: C_D_0               double    input     description = "zero lift drag coef"
% variable: e                   double    input     description = "AVL output: span efficiency"
% variable: user_input_vel_in   double    input     description = "desired run velocity"     units="m/s"
% variable: AspectRatio         double    output    description =  “Wing aspect ratio"  matlabName="AR"
% variable: L_over_D_max        double    output    description = "Maximum Lift/Drag"
% variable: v_stall             double    output    description ="Stall velocity"                units="m/s"
% variable: user_input_vel_out double     output    description ="User velocity passed through"  units="m/s"

% matlab initial values for test
% close all; clear all; clc;
% user_input_vel_in          = 13
% S_in2                      = 111
% b_in                       = 24
% mass                       = 1
% den                        =1.225
```

```
% e                          =0.92
% velo                       =14
% climb_rate                 =0.01
% C_D_0                      =0.1
% C_L_max                    = 1.2

g                             = 9.81
b_m                           = b_in*0.0254
S_m2                          = S_in2*0.0254^2
w                             = mass*g
AR                            = (b_m^2)/S_m2
K                             = 1/(pi*e*AR)

L_over_D_max                 = 1/(sqrt(4*C_D_0*K))
v_stall                      = sqrt(2*w/(den*S_m2*C_L_max))
user_input_vel_out           = user_input_vel_in
```

c. VeloCalcRange.m

Filename: VeloCalcRange.m
Parent program: Model Center and Matlab Plug-in
Description: This file calculates the theoretical max range velocity, max L/D, and stall velocity.
Code as follows:

```
% variable: C_L_max          double   input description = "Max lift coefficient"
% variable: mass             double   input description = "mass"   units="kg"
% variable: S_in2            double   input description = "Planform area"        units="in^2"
% variable: den              double   input description = "Density"              units="kg/m^3"
% variable: b_in             double   input description = "reference span"       units="in"
% variable: C_D_0            double   input description = "zero lift drag coef"
% variable: e                double   input description = "AVL output: span efficiency"
% variable: AspectRatio      double output description = "Wing aspect ratio"  matlabName="AR"
% variable: L_over_D_max     double output description ="Maximum Lift/Drag"
% variable: v_stall          double output description ="Stall velocity"          units="m/s"
% variable: v_range          double output description ="Max range velocity"   units="m/s"

% matlab initial values for test
% close all; clear all; clc;
% user_input_vel_in                 = 13
% S_in2                             = 111
% b_in                              = 24
% mass                              = 1
% den                               =1.225
% e                                 =0.92
% velo                              =14
% climb_rate                        =0.01
% C_D_0                             =0.1
% C_L_max                           = 1.2

g                                   = 9.81
b_m                                 = b_in*0.0254
S_m2                                = S_in2*0.0254^2
w                                   = mass*g
AR                                  = (b_m^2)/S_m2
K                                   = 1/(pi*e*AR)

L_over_D_max                        = 1/(sqrt(4*C_D_0*K))
v_stall                             = sqrt(2*w/(den*S_m2*C_L_max))
v_range                             = sqrt(2*w*(sqrt(K/(C_D_0)))/(den*S_m2))
```

d. VelocitySelection.m

Filename: VelocitySelection.m
Parent Program: Model Center, Matlab Plugin
Description: This script collects values for endurance, max range, user-requested, and stall velocity. It then outputs the use type of flight velocity requested, as well as the lowest possible value above stall for the type of flight requested. Note that the naming of variables at the top of the file can each not go into a second line. Formatting caused it here.
Code as follows:

```
% variable: velocity_setting    double   input        description ="user interface"
           units="1=min_thrust_SC,2=min_pwr_SC,3=user_input_SC,4=user_input"
% variable: C_L_max             double   input        description = "Max lift coefficient"
% variable: mass                double   input        description = "mass"                       units="kg"
% variable: S_in2               double   input        description = "Planform area"              units="in^2"
% variable: den                 double   input        description = "Density"                    units="kg/m^3"
% variable: user_input_vel_in   double   input        description = "desired run velocity"       units="m/s"
% variable: stall_margin        double   input        description = "safety margin above stall" default="0.02"
           matlabName="x"
% variable: v_range             double   input        description = "passed value"              units="m/s"
% variable: v_end               double   input        description = "passed value"               units="m/s"
% variable: run_velocity        double   output       description ="Velocity ultimately selected" units="m/s"
           matlabName="velo"
% variable: v_stall             double   output       description ="Stall velocity"              units="m/s"
% setGroup "StallCheckedValues"
% variable: v_min_power_SC   double output       description ="min power vel, checked above stall"     units="m/s"
% variable: v_min_thrust_SC   double output      description ="min thrust vel, checked above stall"     units="m/s"
% variable: user_input_vel_SC double output      description ="user desired velocity, higher than stall" units="m/s"

% %Uncomment to following inputs for Matlab testing
% S_in2              = 111
% mass               = 1
% den                =1.225
% C_L_max            = 1.16
% x                  = 0.02
% v_range            = 13
% v_end               = 12
% user_input_vel_in=16
% %velocity_setting = 4

% This component allows the user to choose among four different velocities, and checks to see
% if they are above stall. It provides the user a minimum safe velocity, or the use of a desired
% velocity.

% calculate stall velocity
g                    = 9.81
S_m2                 = S_in2*0.0254^2
w                    = mass*g
v_stall              = sqrt(2*w/(den*S_m2*C_L_max))

% Ensure that all velocities are above stall. If not, use value of stall * saftey margin.
% Produce Stall Checked (SC) velocities

if (v_range) < v_stall
  v_min_thrust_SC = (1+x) * v_stall
  else
  v_min_thrust_SC = v_range
end

if (v_end) < v_stall
  v_min_power_SC =  (1+x) * v_stall
```

```
  else
  v_min_power_SC = v_end
end

if (user_input_vel_in) < v_stall
  user_input_vel_SC = (1+x) * v_stall
  else
  user_input_vel_SC =user_input_vel_in
end

% choose which velocity to use based on user's selection
if velocity_setting == 2,
    velo = v_min_power_SC
  elseif velocity_setting == 3,
    velo = user_input_vel_SC
  elseif velocity_setting == 4,
    velo = user_input_vel_in
  else
    velo = v_min_thrust_SC
end

% Now that a run velocity has been selected, it must pass through AVL again to get a new span
% efficiency, which provides a new drag polar K value for further calculations.

% Uncomment below for Matlab testing.
% for velocity_setting=1:4
% if velocity_setting == 2,
%      velo = v_min_power_SC
%    elseif velocity_setting == 3,
%      velo = user_input_vel_SC
%    elseif velocity_setting == 4,
%      velo = user_input_vel_in
%    else
%      velo = v_min_thrust_SC
% end
% end
```

e. AirframeThrustPowerRqd.m

```
Filename: AirframeThrustPowerRqd.m
Parent Program: Model Center
Description: Calculates the airframe thrust and power required.  This was used in the main "Track 1."
Code as follows:
% variable: climb_rate  double     input description ="User prescribed Rate of Climb"       units="m/s"
% variable: den           double    input description ="Density"                             units="kg/m^3"
% variable: C_D_0         double    input description ="AVL input: zero lift drag coef"
% variable: S_in2         double   input description ="Planform area"                        units="in^2"
% variable: b_in          double   input description = "reference span"                      units="in"
% variable: mass          double   input description = "mass"                                units="kg"
% variable: e             double   input description = "AVL output: span efficiency"
% variable: run_velocity     double   input description ="Velocity ultimately selected" units="m/s" matlabName="velo"
% variable: airframeThrustRqd      double     output description ="Thrust required, airframe only"      units="N"
% variable: airframePwrRqd         double    output description ="Power required, airframe only"       units="W"

% Uncomment the following inputs for MatLab testing
% close all; clear all; clc;
% % S_in2              = 111
% b_in                 = 24
% mass                 = 1
```

```
% den                =1.225
% e                  =0.92
% velo               =14
% climb_rate         =0.01
% C_D_0              =0.1

g                    = 9.81
b_m                  = b_in*0.0254
S_m2                 = S_in2*0.0254^2
w                    = mass*g
AR                   = (b_m^2)/S_m2
K                    = 1/(pi*e*AR)

airframeThrustRqd = w*((climb_rate/velo) + 0.5*den*velo^2*(S_m2/w)*C_D_0 + (w*2*K/(S_m2*den*velo^2)))
airframePwrRqd     = velo*airframeThrustRqd
```

f. Thrust_rqd.scriptWrapper

```
Filename: Thrust_rqd.scriptWrapper
Parent program: Model Center
Description:  This script was used in "Track 2."  It calculates the thrust and power required. The only reason why I
wrote it as a Model Center scriptWrapper is because I had not yet incorporated the Matlab plugin.
Code as follows:
# @author: Paul Hrad
# @description: computes thrust required from aero model
# @version: 20 Nov 2009
# declare variables
# Inputs
variable: S_sqin            double input description="Ref planform area (in/sq_in)"     units="in^2"
variable: rho_kgsqm         double input description="density"                          units="kg/m^2"
variable: velocity_ms       double input description="horizontal speed"                 units="m/s"
variable: CD_tot            double input
variable: mass_kg           double input
#variable: gamma_degrees     double input                                                default="1"
variable: climb_rate        double input description="rate of climb(negative if decending)" units="m/s"
variable: altitude_start    double input description="MSL at start"                     units="m"
variable: altitude_end      double input description="MSL at end"                       units="m"
# Outputs
#variable: gamma_rads        double output                                               default="1"
variable: gamma_degs        double output description="climb angle (deg)"               units="deg"
variable: S_sqm             double output description="Ref planform area"               units="m^2"
variable: Thrust_N          double output description="Thrust"                          units="N"
variable: altitude_change   double output description="dela h"                          units="m"
variable: climb_time        double output description="time rqd to climb"               units="sec"
variable: power_rqd_airframe   double output description="airframe power required"      units="W"

#------------------------------------------------
script:
sub run
dim x
altitude_change = altitude_end - altitude_start
if climb_rate = 0 then
  climb_time = 0
  else
  climb_time      = abs(altitude_change / climb_rate)
end if
gamma_rads  = atn(climb_rate / velocity_ms)
gamma_degs  =gamma_rads * 180 / 3.14
S_sqm       = S_sqin*0.00064516
```

```
'Gravity is constant 9.81 N/s^2
x = 0.5*S_sqm*rho_kgsqm*velocity_ms^2*CD_tot + mass_kg*9.81*sin(gamma_rads)
if x > 0 then
  Thrust_N = x
  else
  Thrust_N = 0.001
end if
power_rqd_airframe = Thrust_N * velocity_ms
end sub
```

D.3. Fuel Cell

a. BurnRate.scriptWrapper

Filename: BurnRate.scriptWrapper
Parent program: Model Center
Description: This script file determines the energy usage rate based on a user input efficiency. Nearly any simple Matlab file could be written as a scriptWrapper for Model Center. This code is not used in the model.
Code as follows:

```
#
# @author: Paul Hrad
# @description: Calculates mass loss due to hydrogen expenditure
# @version: 7 Dec 2009
# declare variables
variable: Pelec      double input     description="electrical power required"  units="W"
variable: E          double input     description="fuel specific energy"       units="J/kg"
variable: Rec_eff    double input     description="recovery efficiency"
variable: BurnRate   double output    description="Fuel Burn Rate"             units="Kg/s"
#-------------------------------------------------
script:
Pelec     =1
E         =1
Rec_eff   =1

sub run
BurnRate =Pelec/(Rec_eff*E)
end sub
```

b. FC.m

Filename: FC.m
Parent program: Model Center and Matlab Plug-in
Description: This program first calculates the output current. It creates a polarization plot, and then reads the plot to determine the output current. Finally, it subtracts this amount from the required amount to determine battery power requirement. This is the code related to the model described in this thesis, but was not actually incorporated into the model.
Code as follows:

```
% setGroup Links
% variable: v_input       double      input
% variable: i_input       double      input
% setGroup FC_Voltage_Model
% variable: Eoc           double      input
% variable: A             double      input
% variable: r             double      input
% variable: m             double      input
```

```
% variable: n                double         input
% variable: i_vec            double[]       output
% variable: v_vec            double[]       output
% setGroup FC_Current_Model
% variable: celqty           int            input
% variable: area             double         input
% variable: FC_v             double         output
% variable: act_cur_den      double         output
% variable: act_cur          double         output
% setGroup BatteryInfo
% variable: bat_cur_req      double         output

%This file gives the cell voltage vs. current density plot
% and determines the current produced by PEMFC, 70 deg C.
%close all; clear all; clc;
% -close,clear,clc can cause problems for MC --remove prior to MC
% MC calculated inputs

 v_input   =6; % This value should match the voltage reqd from MC
 i_input   =6;  % Match MC need

format short g
% This voltage model comes from eq 3.14, p. 60 of
% FC Systems Explained, by Larminie
Eoc         =        1.031;
A           =        0.03;         % activation loss constant
r           =        0.000245;
m           =        2.11E-5;
n           =        0.008;
i_vec       =        linspace(1,1000,200);
v_vec       =        Eoc-r*i_vec    - A*log(i_vec)  - m*exp(n*i_vec);

plot(i_vec,v_vec);
xlabel('current (mA / sq cm)')
ylabel('terminal voltage (volts)')

cellqty     =        8;                          % number of cells in stack
area        =        8;                          % active area of FC [cm^2]
FC_v        =        v_input/cellqty;            % per voltage required to match rqd voltage
v_vec_flip=          sort(v_vec,'ascend');       % need to have vector in ascending order
i_vec_flip=          sort(i_vec,'descend');      % to maintain index, flip i vector too.
[volt_index,volt_value] = searchclosest(v_vec_flip,FC_v)
   % Above function searches the voltage vector to find the amount of
   % current density poduced. Search code written by Dr. Murtaza Khan
act_cur_den=         i_vec_flip(volt_index);   % Actual current density needed [mA/cm^2]
act_cur=             act_cur_den*area/1000;    % Actual current produced by FC [amps]

% cur_dens_des=   800; % desired current density [mA/cm^2]
hold on; grid on;
plot(act_cur_den,FC_v,'o')

%The remaining current will need to be produced by the battery
bat_cur_req=i_input-act_cur; %[amps]
```

c. FuelUse.scriptWrapper

Filename: FuelUse.scriptWrapper
Parent Program: Model Center
Description: Uses heating value to calculate loss of mass due to hydrogen burning.

```
Code as follows:
# @author: Paul Hrad
# @description: Calculates mass loss due to hydrogen expenditure
# @version: 7 Dec 2009
# declare variables
variable: Pelec              double input     description="electrical power required"      units="W"
variable: E                  double input     description="fuel specific enthalpy(HHV)"    units="J/kg"
variable: mass               double input     description="mass at start"                  units="kg"
variable: Rec_eff            double input     description="recovery efficiency"
variable: step_duration      double input     description="step size,time"                 units="sec"
variable: burn_rate          double output    description="Fuel Burn Rate"                 units="kg/s"
variable: new_mass           double output
#-----------------------------------------------
script:
' Rec_eff is mean cell voltage/1.48, assuming liquid water as ouput
'          mean cell voltage/1.25, if water vapor is output product
'    Until mean cell voltage is included in model, assume eff=47%
Rec_eff=0.47
' at STP, HHV of hydrogen fuel is 1.43 x 10^8 J/kg
E=143000000
sub run
'calculate the new mass
burn_rate=          Pelec/(Rec_eff*E)
new_mass=           mass-burn_rate*step_duration
end sub
```

D.4. Power and Energy Split Post Processing

a. energy.m

FileName: energy.m
Parent program: Model Center and Matlab Plug-in
Description: Post Processing Component. Calculates the power and energy requirements from a vector of power segment values. Calculates the mass of each power subsystem and calculates aggregate power system mass. Plots and stores mission power profile plot.
Code as follows:

```
% setGroup Profile
% variable: power                      double[]  input
% variable: voltage                    double[] input
% variable: current                    double[] input
% setGroup CreateTimeVector
% variable: climb_rate                 double   input    description ="rate of climb"       units="m/s"
% variable: descend_rate               double   input    description ="rate of descent"     units="m/s"
% variable: leg1_climb_altitude_initial   double   input
% variable: leg1_climb_altitude_final     double   input
% variable: leg1_altitude_change          double   output
% variable: leg2_duration              double   input                                units="min"
% variable: leg3_duration              double   input                                units="min"
% variable: leg4_duration              double   input                                units="min"
% variable: leg5_descend_altitude_initial     double   input
% variable: leg5_descend_altitude_final       double   input
% variable: leg5_altitude_change          double   output
% variable: mission_segment_time        double[] output description="vector of mission segment times" units="sec"
% variable: total_time                  double   output description="Total Mission Time"            units="min"
% setGroup Properties
% variable: FC_specific_power           double   input  description ="fuel cell specific power"     units="W/kg"
% variable: FC_specific_energy       double   input  description ="fuel cell specific energy"     units="W-hr/kg"
```

```
% variable: bat_specific_power         double   input  description ="battery specific power"        units="W/kg"
% variable: bat_specific_energy        double   input  description ="battery specific energy"       units="W-hr/kg"
% setGroup DecisionsResults
% variable: overhead_mass              double   input  description ="weight multiplier for extras"   units="kg"
% variable: differential               double   input  description ="percentage of average for FC size" default="1"
% variable: p_avg                      double   output description ="Average power required"        units="Watts"
% variable: p_fc                       double   output description ="fuel cell power"
% variable: FC_energy                  double   output description ="fuel cell energy"              units="W-hr"
% variable: FC_mass                    double   output description ="weight of fuel cell"           units="kg"
% variable: p_bat                      double   output description ="battery power"
% variable: bat_energy                 double   output description ="battery energy"                units="W-hr"
% variable: bat_mass                   double   output description ="weight of battery"             units="kg"
% variable: bat_use_time_total         double   output description ="Battery time"                  units="hr"
% variable: total_power_mass           double   output description ="overhead*(FC + bat)"           units="kg"
% variable: bat_constrained_by  string  output description ="Which requirement (power or energy) is heavier?"
% variable: fc_constrained_by   string   output description ="Which requirement (power or energy) is heavier?"

% Uncomment the following for Matlab code testing.
% close all; clear all; clc;
% % mission_segment_time    =[60 400 60 120 60]; %secs
% leg1_climb_altitude_initial   =0;
% leg1_climb_altitude_final     =400;
% leg5_descend_altitude_fina  l=0;
% leg5_descend_altitude_initial=400;
% climb_rate                    =1.5;
% leg2_duration                 =3;
% leg3_duration                 =1;
% leg4_duration                 =8;
% descend_rate                  =-.7;
% power                         = [87 92 95 60 50 80 75 60 40];
% differential                 =.50;
% bat_specific_power           =100;
% bat_specific_energy          =90;
% FC_specific_power            =80;
% FC_specific_energy           =100;
% overhead_mass                =1.01;

 leg1_altitude_change          = leg1_climb_altitude_final-leg1_climb_altitude_initial
leg5_altitude_change           = -leg5_descend_altitude_final+leg5_descend_altitude_initial
mission_segment_time(1)        = abs(leg1_altitude_change / climb_rate)
mission_segment_time(2)        = leg2_duration*60 %sec
mission_segment_time(3)        = leg3_duration*60
mission_segment_time(4)        = leg4_duration*60
mission_segment_time(5)        = abs(leg5_altitude_change / descend_rate)
total_time                     = sum(mission_segment_time) /60 %min
t_ms                           = mission_segment_time % mission segement time
p_ms                           = power %mission segment power, length = 9

%   Here, I semi-manually create a bar chart with scatter data.
f = figure('Visible', 'off'); grid off; hold on; box on;
set(gca,'XLim',[0 sum(t_ms)],'YLim',[0 max(p_ms)*1.1])
xaxisdatavalues=[      t_ms(1),          t_ms(1)+t_ms(2),           t_ms(1)+t_ms(2)+t_ms(3),       …
t_ms(1)+t_ms(2)+t_ms(3)+t_ms(4),         ( t_ms(1)+t_ms(2)+t_ms(3)+t_ms(4)+t_ms(5))];%sec
xaxisdatalabels=[round(t_ms(1)/60),round((t_ms(1)+t_ms(2))/60),round((t_ms(1)+t_ms(2)+t_ms(3))/60),…
round((t_ms(1)+t_ms(2)+t_ms(3)+t_ms(4))/60),round((t_ms(1)+t_ms(2)+t_ms(3)+t_ms(4)+t_ms(5))/60)];%min
set(gca,'XTick', xaxisdatavalues, 'Layer','top')%xlim([0,sum(t_ms)])
set(gca,'XTickLabel',xaxisdatalabels)
set(gca,'YGrid','on')
%set(gca,'FontSize',12,'TickDirMode','manual')
set(gca,'FontSize',14)
```

```
ylabel('Power (W)','FontSize',18);
%xlabel([",sprintf('\n'),'Time (min)'],'FontSize',18,'VerticalAlignment','top');
xlabel('Time (min)','FontSize',18,'VerticalAlignment','top');
title('Mission Segment Power Profile','FontSize',18);
colormap(pink(2));

%   Create the segment bars
bar(t_ms(1)*.25,                         p_ms(1),'BarWidth',t_ms(1)/2,'EdgeColor','none')
bar(t_ms(1)*.75,                         p_ms(3),'BarWidth',t_ms(1)/2,'EdgeColor','none')
bar(t_ms(1)+.5*t_ms(2),                  p_ms(4),'BarWidth',t_ms(2),  'EdgeColor','none')
bar(t_ms(1)+t_ms(2)+t_ms(3)/2,           p_ms(5),'BarWidth',t_ms(3),  'EdgeColor','none')
bar(t_ms(1)+t_ms(2)+t_ms(3)+t_ms(4)/2,   p_ms(6),'BarWidth',t_ms(4),  'EdgeColor','none')
bar(t_ms(1)+t_ms(2)+t_ms(3)+t_ms(4)+t_ms(5)/4,  p_ms(7),'BarWidth',t_ms(5)/2,'EdgeColor','none')
bar(t_ms(1)+t_ms(2)+t_ms(3)+t_ms(4)+t_ms(5)*.75,p_ms(9),'BarWidth',t_ms(5)/2,'EdgeColor','none')

%   Space the mission segment bars
bar(                   t_ms(1),max([p_ms(3),p_ms(4)]),'BarWidth',0.007*sum(t_ms),'FaceColor', 'w', 'EdgeColor', 'w')
bar(           t_ms(1)+t_ms(2),max([p_ms(4),p_ms(5)]),'BarWidth',0.007*sum(t_ms),'FaceColor', 'w', 'EdgeColor', 'w')
bar(t_ms(1)+t_ms(2)+t_ms(3),max([p_ms(5),p_ms(6)]),'BarWidth',0.007*sum(t_ms),'FaceColor', 'w', 'EdgeColor', 'w')
bar(t_ms(1)+t_ms(2)+t_ms(3)+t_ms(4),max([p_ms(6),p_ms(7)]),'BarWidth',0.007*sum(t_ms),'FaceColor',…
 'w', 'EdgeColor', 'w')

%   Create the black lines indicating a second measurement in same segment
bar(t_ms(1)/2,p_ms(2),                         'BarWidth',0.007*sum(t_ms),'FaceColor', 'k', 'EdgeColor', 'k')
bar(t_ms(1)+t_ms(2)+t_ms(3)+t_ms(4)+t_ms(5)/2,p_ms(8),'BarWidth',0.007*sum(t_ms),'FaceColor', 'k', …
'EdgeColor', 'k')

%Reduce the power segment vector from 9 measurements to 7 discrete sections
p_seg=[0.5*(p_ms(1)+p_ms(2)), 0.5*(p_ms(2)+p_ms(3)),
p_ms(4),p_ms(5),p_ms(6),0.5*(p_ms(7)+p_ms(8)),0.5*(p_ms(8)+p_ms(9))]
%Increase the time segemnt vector from 5 sections to 7 sections.
t_seg_sec=[t_ms(1)/2, t_ms(1)/2,t_ms(2), t_ms(3),t_ms(4), t_ms(5)/2, t_ms(5)/2]
t_sec=[t_seg_sec(1);
       t_seg_sec(1)+ t_seg_sec(2);
       t_seg_sec(1)+ t_seg_sec(2)+ t_seg_sec(3);
       t_seg_sec(1)+ t_seg_sec(2)+ t_seg_sec(3)+ t_seg_sec(4);
       t_seg_sec(1)+ t_seg_sec(2)+ t_seg_sec(3)+ t_seg_sec(4) + t_seg_sec(5);
       t_seg_sec(1)+ t_seg_sec(2)+ t_seg_sec(3)+ t_seg_sec(4) + t_seg_sec(5)+ t_seg_sec(6);
       t_seg_sec(1)+ t_seg_sec(2)+ t_seg_sec(3)+ t_seg_sec(4) + t_seg_sec(5)+ t_seg_sec(6)+ t_seg_sec(7)]
p_avg=sum((t_seg_sec.*p_seg))/sum(t_seg_sec)
p_fc=differential*p_avg
p_bat=max(p_ms) - p_fc
  if p_bat < 0
     p_bat = 0
  end
p_avg_line =p_avg.*(ones(1,length(t_sec)+1))
p_fc_line  =p_fc .*(ones(1,length(t_sec)+1))
%p_bat_line =p_bat.*(ones(1,length(t_sec)+1))
h1 = gca;
h2 = axes('Position',get(h1,'Position'));

%Make the line the right length
p_fc_line=p_fc_line(1:7)
p_bat_diff= p_seg - p_fc_line       %7 length average value
p_bat_diff_bar=[p_bat_diff(1);p_bat_diff(1);p_bat_diff(2);p_bat_diff(2);...
             p_bat_diff(3);p_bat_diff(3);p_bat_diff(4);p_bat_diff(4);...
             p_bat_diff(5);p_bat_diff(5);p_bat_diff(6);p_bat_diff(6);...
             p_bat_diff(7);p_bat_diff(7)];
t_bar=[0; t_sec(1); t_sec(1); t_sec(2); t_sec(2); t_sec(3); t_sec(3);...
    t_sec(4); t_sec(4); t_sec(5); t_sec(5); t_sec(6); t_sec(6); t_sec(7)];
```

```
%set(h2,'FontSize',12,'YAxisLocation','right','Color','none','XTickLabel',[],'YTickLabel',[])

plot([0; t_sec],p_avg_line,'--',[0; t_sec],[p_fc p_fc_line],'-.',t_bar,p_bat_diff_bar,'LineWidth',3)

set(h2,'YLim',get(h1,'YLim'))
set(h2,'XLim',get(h1,'XLim'))

legend(h2,'Mission Average','Fuel Cell','Battery Segment Avg')
axis('off');set(h2,'FontSize',14)
saveas(f,'powerplot.jpg');
saveas(f,'powerplot.fig');

FC_energy = p_fc * total_time  / 60  % Watt Hours
%Make the line the right length
%p_fc_line=p_fc_line(1:7)
%p_bat_diff= p_seg - p_fc_line        %7 length average value
active_bat=p_bat_diff > 0          %logical: if battery use is positive, 'active'=1
bat_use_time_total=(sum((active_bat).*(t_seg_sec)))/3600 %hours
p_bat_use = (active_bat).*(p_bat_diff) %vector of power values when used
bat_energy=((p_bat_use)*((active_bat).*(t_seg_sec))')/3600 %Watt Hours
%Determine the minimum mass based on power and energy for bat and FC
min_p_bat_mass=p_bat/bat_specific_power          %kg
min_e_bat_mass=bat_energy/bat_specific_energy     %kg
min_p_fc_mass =p_fc/FC_specific_power            %kg
min_e_fc_mass =FC_energy/FC_specific_energy       %kg
%To meet both power and energy rqmts, select the heavier requirement
bat_mass=max([min_p_bat_mass, min_e_bat_mass])    %kg
if min_p_bat_mass > min_e_bat_mass
   bat_constrained_by = 'power_rqmt'
else
   bat_constrained_by = 'energy_rqmt'
end
FC_mass=max([min_p_fc_mass, min_e_fc_mass])       %kg
if min_p_fc_mass > min_e_fc_mass
   fc_constrained_by = 'power_rqmt'
else
   fc_constrained_by = 'energy_rqmt'
end
total_power_mass=overhead_mass*(bat_mass+FC_mass) %kg
%
% %# @author: Paul Hrad
% %# @description: computes climb and descend time, splits up battery and fuel cell use
% %# @version: 27 Jan 2010
```

D.5. QPROP Related Files

a. motor

```
Filename: motor
Parent Program: QPROP
Description: This is the input motor file with the GenMAV motor
Code as follows:
AXI 2808/20 Motor      ! name


 1       ! motor type  (1 = permanent-magnet brushed or brushless DC motor)

 0.105    ! Rmotor (Ohms)
```

```
1.3     ! Io     (Amps)
1490.0     ! Kv     (rpm/Volt)
```

b. motor.template

Filename: motor.template
Parent program: Model Center
Description: Same as for "motor", except this is the file that does not change. Model Center writes to the "motor" file for execution of Model Center variable inputs, but keeps a ".template" file as the original
Codes as follows: See code for "motor".

c. prop

Filename: prop
Parent Program: QPROP
Description: This is the input propeller used for analysis.
Code as follows:

```
9.5x5 Aer CAM !Name  (Data is not correct. It is extrapolated Graupner6x3 data)

 2.0          ! Nblades

 0.50  5.8   ! CL0      CL_a
 -0.3  1.2   ! CLmin   CLmax

 0.028  0.050  0.050  0.5   !  CD0    CD2u  CD2l    CLCD0
 70000   -0.7        !  REref  REexp


 0.0254    0.0254   1.0  !  Rfac   Cfac   Bfac
 0.0         0.0        4.0  !  Radd   Cadd   Badd

#  r    chord    beta
 0.75  0.66    27.5
 1.0    0.69    22.0
 1.5    0.63    15.2
 2.5    0.55    10.2
 3.1    0.44     6.5
 4.2    0.3       4.6
 4.75  0.19     4.2
```

d. prop.template

Filename: prop.template
Parent program: Model Center
Description: Same as for "prop", except this is the file that does not change. Model Center writes to the "prop" file for execution of Model Center variable inputs, but keeps a ".template" file as the original
Codes as follows: See code for "prop"

e. qcon.def

Filename: qcon.def
Parent program: QPROP
Description: Input conditions for air density, viscosity, and speed of sound
Code as follows:

```
1.225      ! rho  kg/m^3
1.78E-5  ! mu   kg/m-s
```

```
340.0     ! a      m/s
```

f. qcon.template

Filename: qcon.template
Parent program: Model Center
Description: Same as for "qcon.def", except this is the file that does not change. Model Center writes to the "qcon.def" file for execution of Model Center variable inputs, but keeps a ".template" file as the original
Codes as follows: See code for "qcon.def"

g. Qprop.dat

Filename: Qprop.dat
Parent program: QPROP
Description: This is the output results file from QPROP that Model Center then retrieves information from. Model Center knows about this program from the ".fileWrapper."
Code as follows:

```
# QPROP Version 1.22
#
# 9.5x5 Aer CAM
#
# AXI 2808/20 Motor
#   0.10500     Rmotor (Ohms)
#    1.3000     Io     (Amps)
#    1490.0     Kv     (rpm/Volt)
#
#   rho =  1.2250     kg/m^3
#   mu  = 0.17800E-04 kg/m-s
#   a   =  340.00     m/s
#
# 1       2      3       4       5       6        7       8     9      10      11      12       13      14      15     16
17         18     19
#
# V(m/s)   rpm     Dbeta     T(N)     Q(N-m)   Pshaft(W)   Volts    Amps    effmot   effprop  adv      CT        CP
DV(m/s)   eff    Pelec      Pprop      cl_avg  cd_avg
# 13.974  7495.     0.000  0.9318     0.3472E-01   27.25      5.735   6.7175   0.7073   0.4779   0.14758  0.3710E-02
0.1146E-02   1.1436   0.3380   38.53      13.02      0.2230  0.4629E-01
#
# radius   chord   beta     Cl      Cd      Re    Mach     effi     effp    Wa(m/s)    Aswirl      adv_wake
  0.0211  0.0172  29.576  -0.2911  0.16959   25578  0.064   1.0565   5.4048   13.52     -1.545      0.1397
  0.0251  0.0175  26.191  -0.2760  0.10729   29153  0.071   1.0557   2.9980   13.47     -1.451      0.1398
  0.0292  0.0173  23.416  -0.2023  0.09110   31991  0.079   1.0411   5.4758   13.56     -1.017      0.1417
  0.0333  0.0168  21.199  -0.1381  0.07991   34156  0.087   1.0281 -11.6295   13.67     -0.6545     0.1435
  0.0373  0.0161  19.474  -0.0757  0.07099   35997  0.095   1.0153  -1.4442   13.80     -0.3374     0.1453
  0.0414  0.0156  18.173  -0.0113  0.06298   38003  0.104   1.0023  -0.2826   13.95     -0.4775E-01  0.1472
  0.0455  0.0153  17.202   0.0544  0.05583   40297  0.113   0.9887   0.1651   14.11      0.2215      0.1493
  0.0495  0.0150  16.458   0.1181  0.04983   42765  0.121   0.9750   0.3939   14.29      0.4664      0.1514
  0.0536  0.0148  15.838   0.1760  0.04513   45237  0.130   0.9619   0.5230   14.47      0.6758      0.1534
  0.0577  0.0146  15.237   0.2231  0.04177   47489  0.139   0.9507   0.5957   14.63      0.8331      0.1552
  0.0617  0.0142  14.552   0.2540  0.03969   49236  0.148   0.9429   0.6302   14.75      0.9191      0.1565
  0.0658  0.0136  13.687   0.2631  0.03891   50152  0.157   0.9402   0.6325   14.80      0.9155      0.1570
  0.0698  0.0129  12.665   0.2530  0.03919   50195  0.166   0.9421   0.6092   14.77      0.8397      0.1566
  0.0739  0.0121  11.612   0.2335  0.04013   49642  0.175   0.9464   0.5709   14.72      0.7361      0.1559
  0.0780  0.0113  10.658   0.2153  0.04122   48864  0.185   0.9506   0.5308   14.66      0.6447      0.1552
  0.0820  0.0106   9.916   0.2079  0.04186   48258  0.194   0.9521   0.5065   14.64      0.5957      0.1550
  0.0861  0.0101   9.392   0.2128  0.04190   47895  0.203   0.9506   0.5025   14.67      0.5893      0.1552
  0.0902  0.0096   9.045   0.2279  0.04149   47660  0.212   0.9464   0.5137   14.73      0.6149      0.1559
```

0.0942	0.0092	8.835	0.2502	0.04088	47409	0.221	0.9402	0.5335	14.83	0.6637	0.1570
0.0983	0.0087	8.718	0.2770	0.04031	46965	0.231	0.9323	0.5556	14.95	0.7281	0.1583
0.1024	0.0082	8.655	0.3050	0.04004	46118	0.240	0.9233	0.5747	15.10	0.8018	0.1598
0.1064	0.0077	8.604	0.3308	0.04033	44629	0.249	0.9135	0.5870	15.25	0.8803	0.1615
0.1105	0.0070	8.529	0.3508	0.04145	42256	0.258	0.9028	0.5890	15.43	0.9659	0.1635
0.1146	0.0062	8.421	0.3609	0.04372	38882	0.268	0.8883	0.5784	15.68	1.090	0.1661
0.1186	0.0053	8.282	0.3412	0.04809	34429	0.277	0.8528	0.5415	16.31	1.446	0.1730

h. qprop.fileWrapper

Filename: qprop.fileWrapper
Parent program: Model Center
Description: The “fileWrapper” file is communicates to Model Center what input and output files and variables interact with the model, how they interact, and where they are.
Code as follows:

```
#
# Basic QPROP filewrapper
#
# @author: Paul Hrad
# @version: 15 May 2009
# @description: Mav QPROP analysis
#
RunCommands
{
# Put ModelCenter values in the input file
 generate inputFile1
 generate inputFile2
 generate inputFile3
 generate inputfile4
# Run the code
 run "qprop_batch.bat"
# Parse the standard output file
 parse outputfile
}
RowFieldInputFile inputFile1
{
templateFile:                qcon.template
fileToGenerate:              qcon.def
setDelimiters "= ,"
setGroup Flight_conditions
variable: rho                double 1 1      description="density"
variable: mu                 double 2 1      description="dynamic viscosity"
variable: a                  double 3 1      description="speed of sound"
}
RowFieldInputFile inputFile2
{
templateFile:                 qprop_batch.template
fileToGenerate:               qprop_batch.bat
setDelimiters "= ,"
variable: velocity           double 1 4      default=1.0     description="flight velocity (m/s)"
variable: RPM                double 1 5
variable: volts               double 1 6
variable: dBeta               double 1 7      default=0       description="beta change (deg)"
variable: Thrust_to_generate  double 1 8      default=1.0     description="thrust required (N)"
variable: torque              double 1 9      default=0       description="(N-m)"
variable: amps                double 1 10     default=0       description="current(Amps)"
variable: Power_Elec          double 1 11     default=0       description="Electrical Power(W)=I*Volts"
}
RowFieldInputFile inputFile3
```

```
{
templateFile:              motor.template
fileToGenerate:            motor
setDelimiters " ,"
setGroup inputs_Motor
variable: Rmotor           double 6 1                description="Rmotor(ohms)"
variable: Io               double 7 1                description="Io (amps)"
variable: Kv               double 8 1                 description="Kv (rpm/volts)"
}
RowFieldInputFile inputFile4
{
templateFile:              prop.template
fileToGenerate:            prop
setDelimiters " ,"
setGroup inputs_Prop
variable: Blade_Number     double 4 1          description="Number of blades"
setGroup radius
variable: r1               double 17 1         description="radius 1 "
variable: r2               double 18 1         description="radius 2 "
variable: r3               double 19 1         description="radius 3 "
variable: r4               double 20 1         description="radius 4 "
variable: r5               double 21 1         description="radius 5 "
variable: r6               double 22 1         description="radius 6 "
variable: r7               double 23 1         description="radius 7 "
setGroup chord
variable: c1               double 17 2         description="chord 1 "
variable: c2               double 18 2         description="chord 2 "
variable: c3               double 19 2         description="chord 3 "
variable: c4               double 20 2         description="chord 4 "
variable: c5               double 21 2         description="chord 5 "
variable: c6               double 22 2         description="chord 6 "
variable: c7               double 23 2         description="chord 7 "
setGroup beta
variable: b1               double 17 3         description="beta 1 "
variable: b2               double 18 3         description="beta 2 "
variable: b3               double 19 3         description="beta 3 "
variable: b4               double 20 3         description="beta 4 "
variable: b5               double 21 3         description="beta 5 "
variable: b6               double 22 3         description="beta 6 "
variable: b7               double 23 3         description="beta 7 "
}
RowFieldOutputFile outputFile
{
# This routine parses the program output file.
# Other variables can be extracted as desired.
fileToParse: Qprop.dat
setDelimiters "= "
markAsBeginning "V(m/s)"
setGroup Results
variable: Velocity          double 2   2      description="Flight Velocity (reflected from input)"
variable: RPM               double 2   3      description="RPM"
variable: dbeta             double 2   4      description="Pitch Change Angle"
variable: Thrust_Propeller  double 2   5      description="Propeller Thrust (N)"
variable: Q                 double 2   6      description="torque"
variable: Pshaft            double 2  7
variable: Voltage           double 2  8
variable: Current           double 2  9
variable: Efficiency_Motor  double 2  10
variable: Efficiency_Propeller        double 2  11
variable: Advance_Ratio               double 2  12
```

```
variable: Thrust_Coefficient          double 2  13
variable: Power_Coefficient           double 2  14
variable: DV                          double 2  15         description="slipstream velocity increment"
variable: Efficiency_Overall          double 2  16
variable: Power_Rqd_Electric          double 2  17
variable: Power_Propeller             double 2  18
variable: cl_avg                      double 2  19
variable: cd_avg                      double 2  20
}
```

i. qprop_batch.bat

Filename: qprop_batch.bat
Parent program: QPROP
Description: Single file is the interface of QPROP for batch mode application. It also tells QPROP to run with a certain set of parameters and then save the results to the ".dat" file.
Code as follows:

```
qprop prop motor 13.41  0 0  0 1 0 0 0  > Qprop.dat 2>&1
REM notes http://web.mit.edu/drela/Public/web/qprop/qprop_doc.txt
REM Below lists the required inputs. In brackets are optional]
REM qprop propfile motorfile Vel Rpm [ Volt dBeta Thrust Torque Amps Pele ]   (single-point)
REM "dBeta", which is the pitch-change angle in degrees, (assumed zero if omitted)
```

j. qprop_batch.template

Filename: qprop_batch.template
Parent program: Model Center
Description: Same as for "qprop_batch.bat", except this is the file that does not change. Model Center writes to the ".bat" file for execution of Model Center variable inputs, but keeps a ".template" file as the original
Codes as follows: See code for "qprop_batch.bat"

D.6. Verification of AVL and QPROP

a. adv_ratio.m

Filename: adv_ratio.m
Parent Program: Matlab
Description: This file takes the values created from QPROP and independently calculates thrust, torque, and power coefficients for plotting. Advance ratio is also calculated.
Code as follows:

```
%calculate and plot prop eff vs. J

close all; clear all; clc;
% First save the data from MC's Data Exporer
% Strip off the first column and blanks for failed runs

data      =load('adv_rat2.csv');
v         =data(1,:);
effp      =data(2,:);
rpm       =data(5,:);
cp        =data(9,:);
ct        =data(10,:);
adv       =data(11,:);
p_prop    =data(12,:);
t         =data(14,:);
```

```
q           =data(7,:);
% Below are the names within the data vector.
% 1         independent variable (Model.System_Variables_Inputs.velocity_user_defined)
% 2         dependent variable (Model.Track1_Primary_Aero.QProp_DragPolar.Results.Efficiency_Propeller)
% 3         dependent variable (Model.Track1_Primary_Aero.QProp_DragPolar.Results.Efficiency_Motor)
% 4         dependent variable (Model.Track1_Primary_Aero.QProp_DragPolar.Results.Efficiency_Overall)
% 5         dependent variable (Model.Track1_Primary_Aero.QProp_DragPolar.Results.RPM)
% 6         dependent variable (Model.Track1_Primary_Aero.QProp_DragPolar.Results.dbeta)
% 7         dependent variable (Model.Track1_Primary_Aero.QProp_DragPolar.Results.Q)
% 8         dependent variable (Model.Track1_Primary_Aero.QProp_DragPolar.Results.Pshaft)
% 9         dependent variable (Model.Track1_Primary_Aero.QProp_DragPolar.Results.Power_Coefficient)
% 10        dependent variable (Model.Track1_Primary_Aero.QProp_DragPolar.Results.Thrust_Coefficient)
% 11        dependent variable (Model.Track1_Primary_Aero.QProp_DragPolar.Results.Advance_Ratio)
% 12        dependent variable (Model.Track1_Primary_Aero.QProp_DragPolar.Results.Power_Propeller)
% 13        dependent variable (Model.Track1_Primary_Aero.QProp_DragPolar.Results.Power_Rqd_Electric)
% 14        dependent variable (Model.Track1_Primary_Aero.QProp_DragPolar.Results.Thrust_Propeller)
% 15        dependent variable (Model.Track1_Primary_Aero.QProp_DragPolar.Results.Voltage)

w           =rpm.*pi./30;
R           =4.75*(0.0254);
den         =1.225;
n           =rpm./60;
d           =2*R;

CT0         =t./(n.^2.*d^4);
CQ0         =q./(n.^2.*d^5);
CP0         =p_prop./(n.^3.*d^5);
J0          =v./(n.*d);
E0          =J0.*CT0./CP0

figure(5)
plot(J0,CT0); grid on;
xlabel('Advance Ratio = V*60/(RPM*D)','FontSize',14)
ylabel('Thrust Coefficient','FontSize',14)
set(gca,'FontSize',14)

figure(6)
plot(J0,CQ0); grid on;
xlabel('Advance Ratio = V*60/(RPM*D)','FontSize',14)
ylabel('Torque Coefficient','FontSize',14)
set(gca,'FontSize',14)

CT          =t./(0.5*den.*(w.*R).^2.*pi.*R^2);
CP          =q./(0.5*den.*(w.*R).^2.*pi.*R^3);
ADV         =v./(w.*R);

figure(1)
plot(ADV,CT,'+r',adv,ct,'b')
hold on;
plot(ADV,CP,'+r',adv,cp,'b')
%title('Advance Ratio vs. Cp and Ct')
grid on;
xlabel('Advance Ratio, w=RPM*pi/30','FontSize',14)
ylabel('QPROP Thrust and Power Coefficient','FontSize',14)
annotation(figure(1),'textbox',...
    [0.410285714285714 0.301428571428573 0.085 0.06],...
    'String',{'Power'},...
    'FitBoxToText','on','FontSize',12);
annotation(figure(1),'textbox',...
    [0.407975446428571 0.757142857142857 0.085 0.06],...
```

```
    'String',{'Thrust'},...
    'FitBoxToText','on','FontSize',12);
set(gca,'FontSize',14)
legend('Recalculated', 'QPROP Direct')

figure(2)
ETA=ADV.*CT./CP;
plot(ADV,ETA,'+r',adv,effp,'b')
set(gca,'FontSize',14,'XLim',[0.1021 0.1406], 'YLim',[0.55 0.575])
legend('Recalculated','QPROP')
xlabel('Advance Ratio, w=RPM*pi/30')
ylabel('Propeller Efficiency')
grid on
```

b. DragPolarPlots.m

Filename: DragPolarPlots.m
Parent Program: Matlab
Description: This file takes output from AVL, determines a drag polar by polynomial fit, and then plots the data
Code as follows:

```
close all; clear all; clc;

% First save the data from MC's Data Exporer
% Strip off the first column and blanks for failed runs
% Strip off CL values higher than 1.16
data=load('CD_CL.csv');
data_chopped=load('CD_CL_2.csv')
v=data(1,:);
CL=data(2,:);
CD=data(3,:);

vc=data_chopped(1,:);
CLc=data_chopped(2,:);
CDc=data_chopped(3,:);

p1 = polyfit(CL.^2,CD,1)
p2 = polyfit(CL,CD,2) %easier form for calculating drag
p3 = polyfit(CD,CL,2) % for ploting traditional drag polar
p1c = polyfit(CLc.^2,CDc,1)
p2c = polyfit(CLc,CDc,2) %easier form for calculating drag
p3c = polyfit(CDc,CLc,2) % for ploting traditional drag polar

% f = figure('Visible', 'on')
% This is a polynomial fit to the entire velocity run from prestall 9.5 to
% 25 m/s.
figure(1)
subplot(1,2,1);
xfit = -1:0.01:6;
yfit = polyval(p1,xfit);
plot(CL.^2,CD,xfit,yfit,'--')
xlabel('CL^2');ylabel('CD');title('Drag Polar')
grid on; legend('Data','Linear Fit','Location','NorthWest');ylim([0,.5])
subplot(1,2,2);
%This is the traditional drag polar presentation
xfit = 0:0.001:0.35;
yfit = polyval(p3,xfit);
plot(CD,CL,xfit,yfit,'--')
%xlim([0 .35]);ylim([0 2]);
xlabel('CD'); ylabel('CL');title('Drag Polar')
```

```
grid on; legend('Data','Quadratic Fit','Location','NorthWest');ylim([0,2])
 % Below is the plot that performs polynomial fit up to stall
figure(2)
subplot(1,2,1);
xfit = -1:0.04:6;
yfit = polyval(p1c,xfit);
plot(xfit,yfit,'-k',CLc.^2,CDc,'.b')%,'LineWidth',2
%set(gca,'xlim',([0.1411 1.393]),'ylim',([0.1121 0.1926]),'fontsize',14)
set(gca,'xlim',([0 1.393]),'ylim',([0.1 0.1926]),'fontsize',14)
% title('Drag Polar Truncated')
xlabel('(Lift Coefficient)^2');ylabel('Drag Coefficient');
grid on; legend('Linear Fit','AVL Data','Location','SouthEast');
 subplot(1,2,2);
%This is the traditional drag polar presentation
xfit = -.5:0.005:0.35;
yfit = polyval(p3c,xfit);
plot(xfit,yfit,'-k',CDc,CLc,'.b')%,'LineWidth',2
%set(gca,'xlim',([0.1117 0.1926]),'ylim',([0.3693 1.18]),'fontsize',14)
set(gca,'xlim',([0.08 0.1926]),'ylim',([0 1.18]),'fontsize',14)

%xlim([0 .35]);ylim([0 2]);
%title('Drag Polar Truncated')
xlabel('Drag Coefficient'); ylabel('Lift Coefficient');
grid on; legend('Quadratic Fit','AVL Data','Location','NorthWest')
```

D.7. Mission Driver Files

a. Send_data_script.txt

```
Filename: Send_data_script.txt
Parent program: Model Center
Description: This handy little script can be used to save a series of data from iterations. It can be useful to save data outside of Model Center's Data Explorer.   I owe credit to J. Simmons for his help in authoring this.
Code as follows:

power.length=9
voltage.length=9
current.length=9
mission_segment_time.length=5
sub run
mission_segment_time(0)=t1
mission_segment_time(1)=t2
mission_segment_time(2)=t3
mission_segment_time(3)=t4
mission_segment_time(4)=t5
'--------
  data = ""
  for i = 0 to 3
    data = data & mission_segment_time(i) & ", "
  next

  data = data & mission_segment_time(4)  ' last element does not need a ','

  f.fromString(data)                              ' copy data from string to file
  f.toFile("C:\Program Files\Phoenix Integration\Analysis Server 6.01\analyses\PostProcessing\timedatafile.csv")
              ' write file to c:\data.csv
 '----------
  data = ""
```

```
 for i = 0 to 7
   data = data & power(i) & ", "
 next

 data = data & power(8)  ' last element does not need a ','

 f.fromString(data)                              ' copy data from string to file
 f.toFile("C:\Program Files\Phoenix Integration\Analysis Server 6.01\analyses\PostProcessing\powerdatafile.csv")
            ' write file to c:\data.csv
 '--------
 data = ""
 for i = 0 to 7
   data = data & voltage(i) & ", "
 next

 data = data & voltage(8)  ' last element does not need a ','

 f.fromString(data)                              ' copy data from string to file
 f.toFile("C:\Program Files\Phoenix Integration\Analysis Server 6.01\analyses\PostProcessing\voltagedatafile.csv")
            ' write file to c:\data.csv
 '----------
 data = ""
 for i = 0 to 7
   data = data & current(i) & ", "
 next

 data = data & current(8)  ' last element does not need a ','

 f.fromString(data)                              ' copy data from string to file
 f.toFile("C:\Program Files\Phoenix Integration\Analysis Server 6.01\analyses\PostProcessing\currentdatafile.csv")
            ' write file to c:\data.csv
end sub
```

b. Mission_Driver

Filename: Mission_Driver
Parent Program: Model Center
Description: This script written within Model Center is a driver that tells Model Center exactly how to run a mission. For each mission certain variables are changed, and other variable values are captured. The Mission_driver accumulates power, voltage, and current for the predesigned mission.
Code as follows:

```
power_record.length  = 9
voltage_record.length= 9
current_record.length= 9
sub run
initial_altitude        = leg1_climb_altitude_initial
final_altitude          = leg1_climb_altitude_final
climb_rate               = leg1_climb_rate
velocity_setting        = leg1_velocity_setting
'initial, midaltitude, and peak climb altitude segment
run_altitude_MSL  =leg1_climb_altitude_initial
power_record(0)     = power
voltage_record(0)   = voltage
current_record(0)   = current
run_altitude_MSL  =((leg1_climb_altitude_final-leg1_climb_altitude_initial)/2)
power_record(1)     = power
voltage_record(1)   = voltage
current_record(1)   = current
run_altitude_MSL  =leg1_climb_altitude_final
```

```
power_record(2)     = power
voltage_record(2)   = voltage
current_record(2)   = current
'"Dash" flight segment
initial_altitude    = leg2_altitude
final_altitude      = leg2_altitude
run_altitude_MSL    = leg2_altitude
climb_rate          = 0
velocity_setting    = leg2_velocity_setting
power_record(3)     = power
voltage_record(3)   = voltage
current_record(3)   = current
'loitering segment
initial_altitude    = leg3_altitude
final_altitude      = leg3_altitude
run_altitude_MSL    = leg3_altitude
velocity_setting    = leg3_velocity_setting
power_record(4)     = power
voltage_record(4)   = voltage
current_record(4)   = current
'"Cruise" flight segment
initial_altitude    = leg4_altitude
final_altitude      = leg4_altitude
run_altitude_MSL    = leg4_altitude
velocity_setting    = leg4_velocity_setting
power_record(5)     = power
voltage_record(5)   = voltage
current_record(5)   = current
'descent from peak to midpoint to ground
run_altitude_MSL    =leg5_descend_altitude_initial
climb_rate          =leg5_climb_rate
velocity_setting    = leg5_velocity_setting
power_record(6)     = power
voltage_record(6)   = voltage
current_record(6)   = current
run_altitude_MSL    =((leg5_descend_altitude_initial-leg5_descend_altitude_final)/2)
power_record(7)     = power
voltage_record(7)   = voltage
current_record(7)   = current
run_altitude_MSL    =leg5_descend_altitude_final
power_record(8)     = power
voltage_record(8)   = voltage
current_record(8)   = current
end sub
```

Bibliography

[1]Department of Defense, "Casualty Update," http://www.defense.gov/NEWS/casualty.pdf, accessed 2 Jan 10, 31 Dec 09.

[2]Dickman, G. G., "Rapid Prototyping of Micro Air Vehicle Control Systems," AIAA 2005-7068, *Infotech@Aerospace Conference*, Arlington VA, 2005.

[3]McConnell, V. P., "Military UAVs Claiming the Skies with Fuel Cell Power," Fuel Cells Bulletin, Dec 07.

[4]Bradley, T., Moffitt, B. A., Parekh, D. E., Fuller, T. F., and Mavris, D. N., "Energy Management for Fuel cell-powered Hybrid-Electric Aircraft," AIAA 2009-4590, *International Energy Conversion Engineering Conference*, Reston VA, 2009.

[5]Office of the Secretary of Defense, U.S. Department of Defense (Acquisitions, Technology, and Logistics), Air Warfare, "Unmanned Aircraft Systems Roadmap 2002-2027," http:// www.acq.osd.mil/usd/UAS_roadmap.pdf, accessed 2 Nov 09, Dec 02.

[6]Headquarters, United States Air Force, "United States Air Force Unmanned Aircraft Systems Flight Plan 2009-2047," 18 May 09.

[7]Office of the Secretary of Defense, "FY2009-2034 Unmanned Systems Integrated Roadmap," http://www.acq.osd.mil/uas/docs/UMSIntegratedRoadmap2009.pdf, accessed 2 Jan 10, 2009.

[8]Keller, J., "Unmanned Vehicle Spending in the 2010 DOD Budget to Reach $5.4 Billion," Military and Aerospace Electronics Online Article, http://mae.pennnet.com/display_article/363553/32/ARTCL/none/EXECW/1/Unmanned-vehicle-spending-in-the-2010-DOD-budget-to-reach-$54-billion/, accessed 2 Jan 10, 28 May 09.

[9]Lucintel, "Opportunities for Composites in the Global UAS Market 2009-2014: Trends, Forecast and Opportunity Analysis," http://www.researchandmarkets.com/reports/1086283, accessed 2 Jan 10, Sep 09.

[10]Mueller, T. J., "On the Birth of Micro Air Vehicles," *International Journal of Micro Air Vehicles*, Vol.1, No.1, 2009.

[11]Boughton, S. E., Attari, T., and Kozak, J., "Comparison and Validation of Micro Air Vehicle Design Methods," AIAA 2004-0406, *AIAA Aerospace Sciences Meeting and Exhibit,* Reston VA, 2004.

[12]Abate, G., "Micro Aerial Vehicle (MAV) Research at AFRL," *2008 International Symposium on Unmanned Aircraft Systems*, Jun 08.

[13]Kohout, L. L., and Schmitz, P. C., “Fuel Cell Propulsion Systems for an All-Electric Air Vehicle,” AIAA 2003-2867, *AIAA/ICAS International Air and Space Symposium and Exposition: The Next 100 Y*, Reston VA, 2003.

[14]Berton, J. J., Freeh, J.E., and Wickenheiser, T.J., “An Analytical Performance Assessment of a Fuel Cell-Powered, Small Electric Airplane,” NASA/TM-2003-212393, 2003.

[15]Moffitt, B. A., Bradley, T. H., Mavris, D., and Parakh, D. E., “Design Space Exploration of Small-Scale PEM Fuel Cell Long Endurance Aircraft,” AIAA 2006-7701, AIAA Aviation Technology, Integration and Operations Conference (ATIO), Reston VA, 2006.

[16] Abate, G., “AFRL Micro Air Vehicle Workshop,” *Micro Air Vehicle (MAV) Workshop*, Dayton OH, Aug 09.

[17]Ofoma, U. C., Wu, C. C., “Design of a Fuel Cell-Powered UAS for Environmental Research,” AIAA 2004-6384, *AIAA “Unmanned Unlimited” Technical Conference*, Reston VA, 2004.

[18]Oates, G. C., *Aerothermodynamics of Gas Turbine and Rocket Propulsion*, American Institute of Aeronautics and Astronautics, Inc., Reston VA, 1997.

[19]Bradley, T. H., Moffitt, B. A., Mavris, D., and Parekh, D. E., “Development and Experimental Characterization of a Fuel Cell-Powered Aircraft,” *Journal of Power Sources*, Vol. 171, 2007.

[20]Wentz, W. H., and Myose, R. Y., “Hydrogen-Fueled General Aviation Airplanes”, AIAA 2005-7324, AIAA *Aviation, Technology, Integration, and Operations Conference (ATIO)*, Reston VA, 2005.

[21]Soban, D. and Upton, E., “Design of a UAS to Optimize Use of Fuel Cell Propulsion Technology,” AIAA 2005-7135, *Infotech@Aerospace*, Reston VA, 2005.

[22]Toensmeier, P., “The Cutting Edge: Fuel Cell Powers Manned Flight,” *Defense Technology International*, Vol. 3, No. 8, Sep 09.

[23]Grasmeyer, J. M., and Keennon, M. T., “Development of the Black Widow Micro Air Vehicle,” AIAA 2001-0127, *AIAA Aerospace Sciences Meeting & Exhibit*, Reston VA, 2001.

[24]Pines, D. J., and Bohorquez, F., “Challenges Facing Future Micro-Air-Vehicle Development,” *Journal of Aircraft*, Vol. 43, No. 2, 2006.

[25]Peterson, B., Erath, B., Henry, K., Lyon, M., Walker, B., Powell, N., Fowkes, K., and Bowman, J., "Development of a Micro Air Vehicle for Maximum Endurance and Minimum Size", AIAA 2003-416, *Aerospace Sciences Meeting and Exhibit*, Reston VA, 2003.

[26]Mueller, T. J., and DeLaurier, J. D., "An Overview of Micro Air Vehicle Aerodynamics," Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Applications, AIAA, Reston VA, 2001.

[27]Babinsky, H., Cattafesta, L., Abate, G., "Design Considerations for a Micro Aerial Vehicle Aerodynamic Characterization Facility at the University of Florida Research and Engineering Education Facility," AIAA 2006-3309, *AIAA Aerodynamic Measurement Technology and Ground Testing Conference*, Reston VA, 2006.

[28]Roberts, C., Vaughan, M., and Bowman, W.J., "Development of a Solar Powered Micro Air Vehicle," AIAA 2002-0703, *Aerospace Sciences Meeting and Exhibit*, Reston VA, 2002.

[29]Rais-Rohani, M., and Hicks, G.R. "Multidisciplinary Design and Prototype Development of a Micro Air Vehicle," *Journal of Aircraft*, Vol. 36, No.1, 1999.

[30]Wikipedia contributors, 'Cooper's Hawk', *Wikipedia, The Free Encyclopedia,* http://en.wikipedia.org/w/index.php?title=Cooper%27s_Hawk&oldid= 309097694, accessed 20 Aug 09.

[31]Tennekes, H., *The Simple Science of Flight: From Insects to Jumbo Jets*, The MIT Press, Cambridge MA, 1996.

[32]Mackenzie, C., "Honor Roll: French Students Develop Innovative UASs," *Defense Technology International*, Vol. 3, No. 11, Dec 09.

[33]Sibilski, K., Pietrucha, J., and Zlocka, M., "The Comparative Evaluation of Power Requirements for Fixed, Rotary, and Flapping Wings Micro Air Vehicles," AIAA 2007-6498, *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, Reston VA, 2007.

[34]Ol, M., Parker, G., Abate, G., and Evers, J., "Flight Controls and Performance Challenges for MAVs in Complex Environments," *AIAA Guidance, Navigation, and Control Conference,* AIAA 2008-6508, Reston VA, 2008.

[35]National Energy Technology Laboratory, *Fuel Cell Handbook*, University Press of the Pacific, 2005.

[36]Anderson, J. D., *Aircraft Performance and Design*, McGraw Hill, Boston, 1999.

[37]Torres, G. and Mueller, T.J., "Micro Aerial Vehicle Development: Design, Components, Fabrication and Flight Testing", *AUVSI unmanned Systems 2000 Symposium and Exhibition*, Reston VA, Jul 00.

[38]US Air Force, "WASP III", *US Air Force Factsheet*, http://www.af.mil/information/factsheets/factsheet_print.asp?fsID=10469&page=1, accessed 11 Jan 10, Nov 09.

[39]Stewart, K., Wagener, J., Abate, G., and Salichon, M., "Design of the Air Force Research Laboratory Micro Aerial Vehicle Research Configuration," AIAA 2007-667, *AIAA Aerospace Sciences Meeting and Exhibit*, Reston VA, 2007.

[40]Harmon, F.G., Frank, A.A., and Chattot, J., "Conceptual Design and Simulation of a Small Hybrid-Electric Unmanned Aircraft System," *Journal of Aircraft*, Vol. 43, No. 5, 2006.

[41]Wikipedia Contributors, "Miniature UAV," Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/w/index.php?title=Miniature_UAV&oldid=340368177, accessed 13 Feb 10.

[42]Chiang, C., Herwerth, C., Mirmirani., M., Ko, A., Matsuyama, S., Choi, S. B., Nomnawee, N., Gamble, D., Arena, A., Gu, G., Wankewycz, T., Koschany, A., and Jin, P., "Systems Integration of a Hybrid PEM Fuel Cell/Battery Powered Endurance UAS", *AIAA 2008-151, AIAA Aerospace Sciences Meeting and Exhibit*, Reston VA, 2008.

[43]Flinn, E. D., "Hydrogen Fuel Cells Power UAS," *Aerospace America*, Nov 06.

[44]Camhi, E. J., Editor., "Endurance Near and (Really) Far," *Aerospace America*, Jun 09.

[45]Bruhn, E. F., *Analysis and Design of Flight Vehicle Structures*, Jacobs Publishing, Indianapolis IN, 1973.

[46]Dufresne, S., Johnson, C., and Mavris, D. N., "Variable Fidelity Conceptual Design Environment for Revolutionary Unmanned Aircraft Systems," *Journal of Aircraft*, Vol. 45, No. 4, 2008.

[47]Bradley, T. H., Moffitt, B. A., Fuller, T. F., and Mavris, D., Parekh, D. E., "Design Studies for Hydrogen Fuel cell-powered Unmanned Aircraft Systems," AIAA 2008-6413, *AIAA Applied Aerodynamics Conference*, Reston VA, 2008.

[48]Gelfi, S., Stefanopoulou A. G., Pukrushpan, J. T. and Peng, H., "Dynamics of low-pressure and high-pressure fuel cell air supply systems," *American Control Conference*, Reston VA, Jun 03.

[49]Moffitt, B. A., Bradley, T. H., Parekh, D. E., and Mavris, D., "Design and Performance Validation of a Fuel Cell Unmanned Aircraft System," AIAA 2006-823, *AIAA Aerospace Sciences Meeting and Exhibit*, Reston VA, 2006.

[50]Moffitt, B. A., Bradley, T. H., Mavris, D., and Parekh, D. E., "Reducing Uncertainty of a Fuel Cell UAS through Variable Fidelity Optimization," AIAA 2007-7793, *AIAA Aviation Technology, Integration and Operations Conference (ATIO)*, Reston VA, 2007.

[51]Goldstein, S., "On the Vortex Theory of Screw Propellers," *Proceedings of the Royal Society of London*, Series A., Vol. 123, No. 792, 1929.

[52]Ogburn, M. J., Nelson, D. J., Wipke, K., and Markel, T., "Modeling and Validation of a Fuel Cell Hybrid Vehicle," SAE 2000-01-1566, *2000 Future Car Congress*, Warrendale PA, 2000.

[53]Lupo, S., Nyberg, H., Karlsson, A., and Mohseni, K., "Xwing – A 3D Viscous Design Tool for Wings," AIAA 2008-173, *AIAA Aerospace Sciences Meeting and Exhibit*, Reston VA, 2008.

[54]Torres, E. T., "Aerodynamics of Low Aspect Ratio Wings at Low Reynolds Numbers with Application to Micro Air Vehicle Design," PhD Thesis, University of Notre Dame IN, 2002.

[55]Khambatta, P., Ukeiley, L., Tinney, C., Stanford, B., and Ifju, P., "Flow Characteristics of a Three-Dimensional Fixed Micro Air vehicle Wing," AIAA 2008-3820, *Fluid Dynamics Conference and Exhibit*, Reston VA, 2008.

[56]McMasters, J. H., and Henderson, M. L., "Low Speed Single Element Airfoil Synthesis," *Technical Soaring*, Vol. 6, No. 2, 1980.

[57]Baxter, D., and East, R., "A Survey of Some Fundamental Issues in Micro-Air-Vehicle Design," *Proceedings of the 14the Bristol International Conference on UAS Systems*, Paper 34, University of Bristol, Bristol, U.K., 1999.

[58]Guglielmo, J. J., and Selig, M. S., "Spanwise Variations in Profile Drag for Airfoils at Low Reynolds Numbers," *Journal of Aircraft*, Vol. 33, No. 4, 1996.

[59]Kunz, P. J., and Kroo, I., "Analysis and Design of Airfoils for Use at Ultra-Low Reynolds Numbers," AIAA, Reston VA, 2001.

[60]Stewart, K., Abate, G., and Evers, J., "Flight Mechanics and Control Issues for Micro Air Vehicles," AIAA-2006-6638, *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, Reston VA, 2006.

[61]Stewart, K. C., Blackburn, K., Wagener, J., Czabaranek, J., and Abate, G., "Development and Initial Flight Tests of a Single-Jointed Articulated-Wing Micro Air Vehicle," AIAA 2008-6708, *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, Reston VA, 2008.

[62]Drela, M., and Youngren, H., "AVL 3.26 User Primer," "avl_doc.txt," http://web.mit.edu/drela/Public/web/avl/avl_doc.txt, accessed 15 Aug 09, 29 Apr 06.

[63]Shields, M. C., Hatcher, C. M., Pitcairn, R. A., Aiken, C. W., Berman, D. B., Carnahan, C. M., Foley, W. A., Hammervold, S. P., Holway, W. E., Marek, L. C, "Design and Development of a Reconnaissance Micro Air Vehicle and Launch System," http://www.colorado.edu/ASEN/SrProjects/Archive/AIAA_Paper_Awards/MARVLIS_AIAA_Final.pdf, accessed 15 Sep 09, 2007.

[64]Blake, W. B., and Simon, J. M., "Tools for Rapid Analysis of Aircraft and Missile Aerodynamics," AIAA 98-2793, Reston VA, 1998.

[65]Selig, M. S., Gopalarathnam, A., Giguere, P., and Lyon, C. A., Systematic Airfoil Design Studies at Low Reynolds Numbers," *Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Applications*, AIAA, Reston VA, 2001.

[66]Fultz, B., "Types of Batteries, and Differences in Energy and Power," Lecture APh 150, *Topics in Applied Physics*, California Institute of Technology, 2007.

[67]Stux, A. M. and Swider-Lyons, K., "Survey of Commercial Small Lithium Polymer Batteries," NRL/MR6110—07-9073, 2007.

[68]Fellner J. P., Miller, R. M., and Shanmugasundaram, V., "Rechargeable Lithium-Ion Based Batteries and Thermal Management for Airborne High Energy Electric Lasers," 06PSC-58, *SAE International*, 2006.

[69]Lu, W., Krysiak, M., and Campbell, D., " Advanced Lithium-Polymer Battery materials for Hybrid Power Systems," Abstract submitted for *44th Power Sources Conference*, Las Vegas NV, Jun 10.

[70] W. L. Gore & Associates, Inc, "Gore Primea Series 56 MEAs, Membrane Electrode Assemblies," Product literature, 2003.

[71] Woodbank Communications, Ltd., Ragone Plot, http://www.mpoweruk.com/performance.htm, accessed 13 Feb 10.

[72]Mueller, T. J., Kellogg, J. C., Ifju, P. J., and Shkarayev, S. V., *Introduction to the Design of Fixed-Wing Micro Air Vehicles*, American Institute of Aeronautics and Astronautics, Inc., Reston VA, 2007.

[73]Grove, W. R., "On Voltaic Series and the Combination of Gases by Platinum," *Philosophical Magazine and Journal of Science,* Vol. XIV, 1839.

[74]Warshay, M. and Prokopius, P.R., "The Fuel Cell In Space," NASA TM 102366, http://74.125.113.132/search?q=cache:LnVi0pk0m5oJ:ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19900002488_1990002488.pdf+fuel+cell+space+program&cd=3&hl=en&ct=clnk&gl=us, accessed 15 Oct 09, 1989.

[75]Larminie, J., and Dicks, A., *Fuel Cell Systems Explained*, 2nd Edition, Wiley, New York, 2003.

[76]Chang, H., "Power Performance and Environmental Reliability of 25W DMFC System for Soldier Power Application," abstract submitted for the *12th Annual Small Fuel Cells International Conference*, Cambridge MA, Apr 10.

[77]Manwell, J. F. and McGowan, J. G., "Lead Acid Battery Storage Model for Hybrid Energy Systems," *Solar Energy*, 1993.

[78]Liaw B. Y., Nagasubrmanian, G., Jungst, R. G., and Doughty, D. H., "Modeling of Lithium Ion Cells—A Simple Equivalent-Circuit Model Approach," *Solid State Ionics*, Vol 175, 2004.

[79]Gomadam, P. M., Merritt, D. R., Scott, E. R., Schmidt, C. L., Skarstad, P. M., and Weidner, J. W., "Modeling Li/CFx-SVO Hybrid-Cathode Batteries," *Journal of The Electrochemical Society*, Vol 154, 2007.

[80]Mattingly, J.D., Heiser, W.H., Pratt, D.T., *Aircraft Engine Design*, American Institute of Aeronautics, and Astronautics, Reston VA, 2002.

[81]Drela, M., and Youngren, H., "*QPROP*," Software Package, Ver 1.22, http://web.mit.edu/drela/Public/web/QPROP/, accessed 15 Aug 09.

[82]Drela, M., "QPROP Users Guide," http://web.mit.edu/drela/Public/web/QPROP/QPROP_doc.txt , accessed 15 Aug 09, 6 Jul 07.

[83]Anderson, J. D., *Aircraft Performance and Design*, McGraw-Hill, Boston, 1999.

[84]Ostler, J. N., Bowman, W. J., Snyder, D. O., and McLain, T. W., "Performance Flight Testing of Small, Electric Powered Unmanned Aerial Vehicles," *International Journal of Micro Air Vehicles*, Vol. 1, No. 3, 2009.

[85]Lundstrom, D., Amadori, K., and Krus, P., "Validation of Small Scale Electric Propulsion System Models," AIAA 2010-483, *AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, Reston VA, Jan 10.

[86]Beam, J., Heidenreich, J., Miller, R., Fellner, J., Hager, B., "Development of a Distributed Integrated Modeling Environment to Study the Impact of Subsystem Performance on an Air Vehicle Design," AIAA 2005-5573, *International Energy Conversion Engineering Conference*, Reston VA, 2005.

[87]Beam, J., Miller, R., and Heidenreich, J., "HAVE GOLD III, Development of an Analysis Modeling Environment Interim Report," Schafer Corporation, 2006.

[88]Turan, M., "Tools For the Conceptual Design and Engineering Analysis of Micro Air Vehicles," MS Thesis, College of Aeronautical Engineering, Air Force Institute of Technology, Wright Patterson Air Force Base OH, 2009.

[89]Schilter, A. L., McKay, D. A., and Stefanopoulou, A. G., "Parameterization of Fuel Cell Stack Voltage: Issues on Sensitivity, Cell-to-Cell Variation, and Transient Response," *4th International Conference on Fuel Cell Science, Engineering, and Technology*, Irvine, CA 2006.

[90]Drela, M., and Youngren, H., "*XFOIL*," Software Package, http://web.mit.edu/drela/Public/web/xfoil/, accessed 15 Aug 09.

[91]Raymer, D. P., *Aircraft Design: A Conceptual Approach*, 4th ed., AIAA, Reston VA, 2006.

[92]Drela, M., and Youngren, H., "*Athena Vortex Lattice (AVL)*," Software Package, Ver 3.26, http://web.mit.edu/drela/Public/web/avl/, accessed 15 Aug 09.

[93] Ol, M., and Zeune, C., "Analytical – Experimental Comparison for Small Electric Unmanned Air Vehicle Propellers," *AIAA 2008-7345, AIAA Applied Aerodynamics Conference*, Reston VA, Aug 08.

[94]Short, S. R., "Modeling and Analysis of Active Turbulators on Low Reynolds Number Unmanned Aerial Vehicles," Masters Thesis, University of Illinois at Urbana-Champaign, Department of Aerospace Engineering, Urbana IL, 2006.

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 074-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to an penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
25-03-2010	Master's Thesis	02-09-08 – 25-03-10

4. TITLE AND SUBTITLE Conceptual Design Tool For Fuel-Cell Powered Micro Air Vehicles	**5a. CONTRACT NUMBER** **5b. GRANT NUMBER** **5c. PROGRAM ELEMENT NUMBER**
6. AUTHOR(S) Paul M. Hrad, Capt., USAF	**5d. PROJECT NUMBER** **5e. TASK NUMBER** **5f. WORK UNIT NUMBER**
7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765	**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT/GAE/ENY/10-M12
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Attn: Dr. Thomas Reitz Air Force Research Laboratory, Propulsion Directorate 1950 Fifth Street, Bldg. 18G, Rm G33 Wright Patterson AFB, OH 45433-7251 DSN 785-4275, Thomas.Reitz@wpafb.af.mil	**10. SPONSOR/MONITOR'S ACRONYM(S)** AFRL/RZPS **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
A conceptual design tool was built to explore power requirements of a hybrid-power system for Micro Air Vehicles (MAVs) comparable in size to the Cooper's Hawk. An inviscid aerodynamic code, Athena Vortex Lattice (AVL), and a motor-propeller analysis code, QPROP, provide overall lift, drag, and thrust data for power-required calculation as functions of many variables to include mass, platform geometry, altitude, velocity, and mission duration. Phoenix Technologies' Model Center was used to integrate multi-disciplinary components that employ specific power and specific energy of two power sources to determine power system mass required for a designated mission. The tool simulated a mission for the fixed wing Generic Micro Aerial Vehicle (GenMAV), and relative sizing between a high specific power source and a high specific energy source was investigated. Current small fuel cell technology provides inadequate specific power. It was found that a MAV-sized fuel cell-battery hybrid-power system would not perform better than a pure battery or battery-battery power system. A feasible fuel cell capability requirement of at least 325 W/kg matched with at least 921 W-hr/kg was identified as a fuel cell – Li-Po solution for a defined 30 min mission resulting in reduced power system mass compared to using only Li-Po batteries.

15. SUBJECT TERMS
Generic Miro Air Vehicle (GenMAV), Power, Athena Vortex Lattice (AVL), QPROP, Model Center

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Frederick G. Harmon, Lt Col, USAF
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** (937) 255-3636, ext 7478 frederick.harmon@afit.edu
UU	UU	UU	UU	232	
					Standard Form 298 (Rev. 8-98) Prescribed by ANSI Std. Z39-18
					Form Approved OMB No. 074-0188